

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2001-189934  
(P2001-189934A)

(43) 公開日 平成13年7月10日 (2001.7.10)

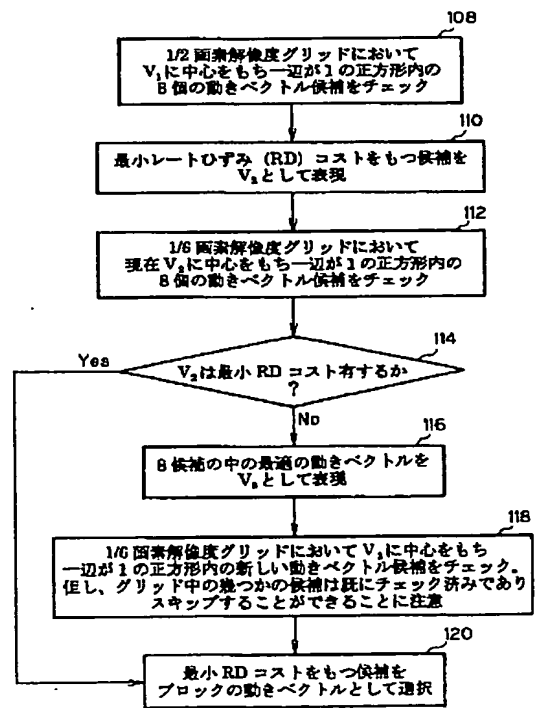
(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード (参考)
H 0 4 N 7/24		H 0 3 M 7/36	
H 0 3 M 7/36		H 0 4 N 7/13	Z

審査請求 未請求 請求項の数22 O L 外国語出願 (全 52 頁)

(21) 出願番号	特願2000-256484(P2000-256484)	(71) 出願人	000005049 シャープ株式会社 大阪府大阪市阿倍野区長池町22番22号
(22) 出願日	平成12年7月24日 (2000.7.24)	(72) 発明者	ジョディー リバス アメリカ合衆国, 98683 ワシントン州, カマス, 16604 エスイー フィッシャー ドライブ
(31) 優先権主張番号	60/146102	(72) 発明者	ジャンドン シェン アメリカ合衆国, 60616 イリノイ州, シ カゴ, 3250 エス. ウエルズ
(32) 優先日	平成11年7月27日 (1999.7.27)	(74) 代理人	100079843 弁理士 高野 明近 (外2名)
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	09/615, 791		
(32) 優先日	平成12年7月13日 (2000.7.13)		
(33) 優先権主張国	米国 (US)		

(54) 【発明の名称】 適応性のある動き精度をもった動き推定方法

(57) 【要約】  
【課題】 計算量の増加が少ない高画素精度の動きベクトルの計算技法を含む適応性のある動き精度をもった動き推定方法を提供する。  
【解決手段】 1つの技法は、最適動きベクトルをスマートに探索する部分画素空間におけるファストサーチ戦略を用いる。別の技法は、異なる段階で異なる補間フィルタを用い高精度動きベクトルを推定して、計算の複雑さを軽減する。他の技法は、異なる動き精度に従って適応するレートひずみ判定基準を用いて最適の動きベクトルと最適の動き精度の両方を決定する。さらに他の技法は、関連動きベクトル精度に従い異なる符号化単位において異なって解釈されるVLCテーブルを使用する。



## 【特許請求の範囲】

【請求項1】 1つのマクロブロックに対する最適な動きベクトルを見つけ出すことにより、動き補償動画符号化における動きベクトルを推定するための、ファストサーチ適応性のある動き精度をもったサーチ方法であつて、

(a) 最適動きベクトル $V_2$ を見つけ出すために、部分画素解像度グリッド上で $V_1$ を中心とする予め定められた正方形範囲内の動きベクトル候補の第1セットをサーチするステップと、(b) 最適動きベクトル $V_3$ を見つけ出すために、部分画素解像度グリッド上で $V_2$ を中心とする予め定められた正方形範囲内の動きベクトル候補の第2セットをサーチするステップと、

(c) 前記マクロブロックの前記最適な動きベクトルを見つけ出すために、部分画素解像度グリッド上で $V_3$ を中心とする予め定められた正方形範囲内の動きベクトル候補の第3セットをサーチするステップとを含んで成ることを特徴とする動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項2】 最適動きベクトル $V_2$ を見つけ出すために、部分画素解像度グリッド上で $V_1$ を中心とする予め定められた正方形範囲内の第1セットの動きベクトル候補をサーチする前記ステップは、最適動きベクトル $V_2$ を見つけ出すために、 $1/2$ 画素解像度グリッド上で $V_1$ を中心とする正方形範囲1内の8個の動きベクトル候補の第1セットをサーチするステップをさらに含むことを特徴とする請求項1に記載の動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項3】 最適動きベクトル $V_3$ を見つけ出すために、部分画素解像度グリッド上で $V_2$ を中心とする予め定められた正方形範囲内の第2セットの動きベクトル候補をサーチする前記ステップは、最適動きベクトル $V_3$ を見つけ出すために、 $1/6$ 画素解像度グリッド上で $V_2$ を中心とする正方形範囲1内の8個の動きベクトル候補の第2セットをサーチするステップをさらに含むことを特徴とする請求項1に記載の動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項4】  $V_2$ が最小レート歪みコストを有する場合に前記ブロックに対する動きベクトルとして $V_2$ を使用するステップと、請求項1のステップ(c)をスキップするステップとをさらに含むことを特徴とする請求項1に記載の動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項5】 前記マクロブロックの前記最適な動きベクトルを見つけ出すために、部分画素解像度グリッド上で $V_3$ を中心とする予め定められた正方形範囲内の第3セットの動きベクトル候補をサーチする前記ステップは、前記マクロブロックの前記最適な動きベクトルを見つけ出すために、 $1/6$ 画素解像度グリッド上で $V_3$ を

中心とする正方形範囲1内の8個の動きベクトル候補の第3セットをサーチするステップをさらに含むことを特徴とする請求項1に記載の動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項6】 前記マクロブロックの前記最適な動きベクトルを見つけ出すために、部分画素解像度グリッド上で $V_3$ を中心とする予め定められた正方形範囲内の第3セットの動きベクトル候補をサーチする前記ステップは、前記第3セットの動きベクトル候補中の既にテスト済みの候補をスキップするステップをさらに含むことを特徴とする請求項1に記載の動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項7】 前記第1セットの動きベクトル候補をサーチする前記ステップは、第1の補間を行うために第1のフィルタを使用して前記第1セットの動きベクトル候補をサーチするステップをさらに含み、前記第2セットの動きベクトル候補をサーチする前記ステップは、第2の補間を行うために第2のフィルタを使用して前記第2セットの動きベクトル候補をサーチするステップをさらに含み、前記第3セットの動きベクトル候補をサーチする前記ステップは、第3の補間を行うために第3のフィルタを使用して前記第3セットの動きベクトル候補をサーチするステップをさらに含むことを特徴とする請求項1に記載の動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項8】 最適動きベクトル $V_3$ を見つけ出すために、部分画素解像度グリッド上で $V_2$ を中心とする予め定められた正方形範囲内の第2セットの動きベクトル候補をサーチする前記ステップは、(a)  $V_2$ が中心にある場合、 $1/3$ ペル精度の3候補 $V_2$ と次に最低のRDコストをもつ $1/2$ ペル位置をサーチするステップと、(b)  $V_2$ がコーナーのベクトルである場合は、 $V_2$ に最も近い $1/3$ ペル精度の4ベクトル候補をサーチするステップと、(c)  $V_2$ が2つのコーナーベクトル間にある場合は、2つのコーナーの中のより低いRDコストをもつコーナーを決定し、より低いRDコストを持つ前記コーナーとの間のラインに最も近い $1/3$ ペル精度の4ベクトル候補をサーチするステップとをさらに含むことを特徴とする請求項1に記載の動きベクトルを推定するためのファストサーチ適応性のある動き精度をもったサーチ方法。

【請求項9】 1つのマクロブロックに対する最適な動きベクトルを見つけ出すことにより、動き補償動画符号化における動きベクトルを推定するための、適応性のある動き精度をもったサーチ方法であつて、(a) 第1補間を行うための第1フィルタを用いて、最適動きベクトル $V_2$ を見つけ出すために、 $V_1$ を中心とするグリッド内の第1セットの動きベクトル候補をサーチするステッ

ブと、(b) 第2補間を行うための第2フィルタを用いて、最適動きベクトル $V_3$ を見つけ出すために、 $V_2$ を中心とするグリッド内の第2セットの動きベクトル候補をサーチするステップと、(c) 第3補間を行うための第3フィルタを用いて、前記マクロブロックの前記最適動きベクトルを見つけ出すために、 $V_3$ を中心とするグリッド内の第3セットの動きベクトル候補をサーチするステップとを含んで成ることを特徴とする動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項10】 第1補間を行うための第1フィルタを用いてサーチする前記ステップは、粗い補間を行うための簡単なフィルタの使用をさらに含むことを特徴とする請求項9に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項11】 第1補間を行うための第1フィルタを用いてサーチする前記ステップは、粗い補間を行うための簡単なフィルタの使用をさらに含み、第2補間を行うための第2フィルタを用いてサーチする前記ステップは、細かい補間を行うための複合フィルタの使用をさらに含むことを特徴とする請求項9に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項12】 第3補間を行うための第3フィルタを用いてサーチする前記ステップは、細かい補間を行うための複合フィルタの使用をさらに含むことを特徴とする請求項11に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項13】 第1補間を行うための第1フィルタを用いてサーチする前記ステップは、 $2 \times 2$ の参照フレームを補間するための双線形フィルタの使用をさらに含むことを特徴とする請求項9に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項14】 第1補間を行うための第1フィルタを用いてサーチする前記ステップは、 $2 \times 2$ の参照フレームを補間するために双線形フィルタの使用をさらに含み、第2補間を行うための第2フィルタを用いてサーチする前記ステップは、細かい補間を行うための立方体フィルタの使用をさらに含むことを特徴とする請求項9に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項15】 第3補間を行うための第3フィルタを用いてサーチする前記ステップは、細かい補間を行うための立方体フィルタの使用をさらに含むことを特徴とする請求項14に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項16】 1つのマクロブロックに対する最適な動きベクトルを見つけ出すことにより、動き補償動画符号化における動きベクトルを推定するための、適応性のある動き精度をもったサーチ方法において、(a) 前

記マクロブロックの第1の最適動きベクトルに対する第1の動き精度をサーチするステップと、(b) 前記第1の最適動きベクトルと前記第1の動き精度を符号化するステップと(c) 少なくとも1つの第2の動き精度で前記マクロブロックの少なくとも1つの第2の最適動きベクトルをサーチするステップと、(d) 前記少なくとも1つの第2の最適動きベクトルと、前記少なくとも1つの第2の動き精度とを符号化するステップと、

(e) 前記第1と少なくとも1つの最適動きベクトルの中から前記最適動きベクトルをレート歪み判定基準を用いて選択するステップとを含んで成ることを特徴とする動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項17】 レート歪み判定基準を用いて前記最適な動きベクトルを選択する前記ステップは、前記最適動きベクトルと前記最適動き精度の両方を決定するために異なる動き精度に従って前記レート歪み判定基準を適応させるステップをさらに含むことを特徴とする請求項16に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項18】 少なくとも1つの第2の動き精度で少なくとも1つの第2の最適動きベクトルをサーチする前記ステップは、前記第1の動き精度より細かい少なくとも1つの第2の動き精度で、少なくとも1つの第2の最適動きベクトルをサーチするステップをさらに含むことを特徴とする請求項16に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項19】 レート歪み判定基準を用いて前記最適な動きベクトルを選択する前記ステップは、前記最適な動きベクトルを選択するために“歪み+ $L \times$ ビット”型のレート歪み判定基準を使用するステップをさらに含むことを特徴とする請求項16に記載の動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項20】 1つのマクロブロックに対する最適な動きベクトルを見つけ出すことにより、動き補償動画符号化における動きベクトルを推定するための適応性のある動き精度をもったサーチ方法であって、(a) 前記マクロブロックの1つの最適動きベクトルに対する1つの動き精度をサーチするステップと、(b) 関連する動きベクトル精度により異なる符号化単位において異なって解釈された可変長符号(VLC)テーブルからのコードを用いて、前記動き精度を符号化するステップと

(c) 個々の精度空間において前記最適な動きベクトルを符号化するステップとを含んで成ることを特徴とする動きベクトルを推定するための適応性のある動き精度をもったサーチ方法。

【請求項21】 1つのマクロブロックに対する1つの最適な動きベクトルを見つけ出すことにより、動き補償動画符号化における動きベクトルを推定するためのシステムであって、(a) 最適動きベクトル $V_2$ を見つけ

出すために、部分画素解像度グリッド上で $V_1$ を中心とする予め定められた正方形範囲内の第1セットの動きベクトル候補をサーチするための第1符号化器と、(b)

最適動きベクトル $V_3$ を見つけ出すために、部分画素解像度グリッド上で $V_2$ を中心とする予め定められた正方形範囲内の第2セットの動きベクトル候補をサーチするための第2符号化器と、(c) 前記マクロブロックの前記最適な動きベクトルを見つけ出すために、部分画素解像度グリッド上で $V_3$ を中心とする予め定められた正方形範囲内の第3セットの動きベクトル候補をサーチするための第3符号化器とを含んで成ることを特徴とする動きベクトルを推定するためのシステム。

【請求項22】 前記第1符号化器、第2符号化器および第3符号化器が単一の符号化器であることを特徴とする請求項21に記載の動きベクトルを推定するためのシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、デジタル動画のビット圧縮又は符号化方法に関し、特に、動き補償動画符号化における動きベクトルを推定して符号化する効果的な方法に関する。

【0002】

【従来の技術】従来の動き推定方法において、符号化する現フレームを、“マクロブロック”と称する、例えば $16 \times 16$ ピクセルの同一サイズの画像ブロックに分割する。各現マクロブロック毎に、符号化器は、現マクロブロックに最も一致する、前に符号化されたフレーム (“参照フレーム”) 内ブロックをサーチする。1個の現マクロブロックと参照フレーム内の最も一致するマクロブロック間の座標移動量は、マクロブロックの2次元ベクトル (“動きベクトル”) によって表現される。動きベクトルの各成分は画素単位で測定される。

【0003】例えば、現マクロブロックに最も一致する参照マクロブロックが同一位置にある場合、静止背景画像が代表的な例であるように、現マクロブロックの動きベクトルは $(0, 0)$ である。最も一致する参照マクロブロックが、現マクロブロックの座標から右に2画素及び上に3画素の位置で発見された場合の動きベクトルは $(2, 3)$ である。かような動きベクトルは、水平成分 $X$ 及び垂直成分 $Y$ が整数画素値を有しているので、整数画素 (又は “整数ペル” 又は “完全ペル”) の精度を有すると云う。図1において、ベクトル $V_1 = (1, 1)$ は、任意の現マクロブロックに対する完全ペルの動きベクトルを表わしている。

【0004】動画シーン中の移動オブジェクトは、しかしながら、フレームからフレームへ整数画素増分単位では移動しない。真の動きは、 $X$ 及び $Y$ 方向に沿った実数値を取る。従って、現マクロブロックにより良く一致する参照マクロブロックは、前フレームを $N \times N$ の因子

(ファクタ) で補間し、次に補間したフレーム内の最も一致するマクロブロックをサーチすることにより発見できることがよくある。動きベクトルは $X$ と $Y$ に沿って $1/N$ 画素の増分値を取ることができ、 $1/N$ 画素 (又は  $1/N$ ペル) の精度を有すると表現される。

【0005】ITU電気通信標準化部門の “H. 26L 規格に対する提案呼びかけへの応答 (Response to Call for Proposals for H.26L)” (Q. 15/SG16, 文書Q15-F-11, ソウル, 1998年11月) 及びITU電気通信標準化部門の “H. 26L 規格に対するTelenorの提案の強化 (Enhancement of theTelenor Proposal for H.26L)” (Q. 15/SG16, 文書Q15-G-25, モントレー, 1999年2月) において、Gisle Bjoitegaardは、動画符号化規格H. 26L (“Telenor符号化器”) に関し $1/3$ ペル精度の動きベクトルと立方体状補間法の使用を提案した。これを実施するために、Telenor符号化器は、立方体状補間フィルタを用いて参照フレームを $3 \times 3$ 補間するか又は “アップサンプリング” する。この補間バージョンは、参照フレームの9倍のメモリを必要とする。1個の与えられたマクロブロックにおいて、Telenor符号化器は最適動きベクトルを2つのステップで推定する。即ち、符号化器は、先ず最適な整数ペルベクトルを探索し、次に $V_1$ に近い最適な $1/3$ 画素精度のベクトル $V_{1/3}$ を探索する。図1の例では、 $3 \times 3$ 補間参照フレーム内の ( $16 \times 16$ 画素) の全8ブロックを調べ、図示のように動きベクトル $V_{1/3} = (VX, VY) = (1 + 1/3, 1)$ に関連するブロックである最も一致するブロックを見つけ出す。このTelenor符号化器は、幾つかの問題を有している。先ず、 $1/3$ ペル精度の動きベクトルを計算するために部分最適なファストサーチ戦略と複雑な立方体フィルタを (全段階において) 使用することである。結果として、計算した動きベクトルは、最適ではなく、莫大な記憶容量と計算量を要し非常に高価につく。さらに、このTelenor符号化器は、 $1/3$ 画素に固定した有効レート歪み判定基準による精度を用い、そのために、より良い動き精度を選択するようには適応できない。同様に、このTelenor符号化器の可変長符号 (“VLC”) テーブルは、 $1/3$ 画素に固定した精度を有しているので、異なる精度に対して適応できず異なる解釈ができない。

【0006】最も良く知られている動画圧縮方法では、 $1/2$ 画素精度で動きベクトルを推定して符号化するが、その理由は、従前の研究によれば、より高い又はより適応性を有する動き精度では計算が複雑になるだけで追加の圧縮利得が得られないことを示唆していたからである。しかしながら、これらの従前の研究は、最適化レート歪み判定基準を用いて動きベクトルを推定しておらず、計算の複雑化を減少させるためにかような基準の凸

状特性を利用しておらず、動きベクトルと動きベクトルの精度を符号化する有効な戦略を用いていなかった。

【0007】かような従前の研究の1つとして、Bernard Girodの論文（以後Girodの論文と略記する）“分数ペル精度による動き補償予測（Motion-Compensating Prediction with Fractional-Pel Accuracy）”（IEEE 通信会報，第41巻4号，604-612頁，1993年4月）が挙げられる。このGirodの論文は、動画符号化に部分画素（サブピクセル）の動き精度を用いる利点に関する最初の基本的な分析である。Girodは、部分画素空間における最適動きベクトルを探索するために簡単な階層戦略を用いた。彼はまた所与の精度に対する最適動きベクトルを選択するために単純平均絶対差分（“MAD”）の基準を用いた。この最適精度は、理想化された仮定に基づくために非実用的な公式を用いて選択され、非常に複雑であり、また、全ての動きベクトルは1フレーム内の同一精度を持つものに制限される。最後に、Girodは、予測誤差エネルギーにのみ注目し、動きベクトルを符号化するためのビットの使用法には言及しなかった。

【0008】他の従前の研究として、Smita Gupta及びAllen Gershoの論文（以後“Guptaの論文”と略記する）“分数画素の動き推定（On Fractional Pixel Motion Estimation）”（SPIE VCIP会報，2094巻408-419頁，ケンブリッジ，1993年11月）が挙げられる。このGuptaの論文は、動画圧縮のために部分画素精度で動きベクトルを計算し選択して符号化する方法を提示している。Guptaの論文は、平均自乗誤差（“MSE”）と双線形補間に基づく公式を開示し、この公式を用いて理想的な動きベクトルを見つけ出し、かようなベクトルを所望の動き精度に量子化した。所与の精度に対する、この最適動きベクトルは、部分最適MSE判定基準を用いて決定され、この最適精度は歪みビット当たりエネルギー差を最大限減少させる方法を用いて選択された。これは、渴望された（部分最適な）判定基準である。所与の動きベクトルの符号化は、最初に1/2ペル精度で符号化し、次に詳細化ビットでより高精度に符号化する方法で行われた。粗から微細への符号化は、かなりのビットを必要とするきらいがある。

【0009】論文“ブロックに基づく動き補償動画符号化器の場合の最適動きベクトル精度（On the Optimal Motion Vector Accuracy for Block-Based Motion-Compensated Video Coders）”（IST/SPIE デジタル動画の圧縮会報：アルゴリズムと技術，302-314頁，サンジョセ，1996年2月）（以後、Ribasの論文と略記する）において、Jordi Ribas-Corbera及びDavid L. Neuhoffは、動き精度のビットレートへの影響をモデル化し、ビットレートを最小にする最適精度を推定する幾つ

かの方法を提案した。このRibasの論文は、任意精度に対する動きベクトルを計算するためのフルサーチ方法を記述し、双線形補間のみを考慮した。最適な動きベクトルはMSEを最小にすることにより発見し、最適精度はレート歪みの最適化により導出された幾つかの公式を用いて選択した。動きベクトルと精度は、リアルタイム装置には実装が難しい、複雑なフレーム適応エントロピー符号化器を用いて符号化した。

【0010】論文“より高いビットレートでの予測強化に関する新しいコア実験の提案（Proposal for a new core experiment on Prediction enhancement at higher bitrates）”（ISO/IEC JTC1/SC29/WG11 動画及び音声の符号化、MPEG97/1827，セビリア，1997年2月）及び“1/4ペル動き補償のための複雑さを軽減した実装装置の性能評価

（Performance Evaluation of a Reduced Complexity Implementation for Quarter Pel Motion Compensation）”（ISO/IEC JTC1/SC29/WG11 動画及び音声の符号化、MPEG97/3146，サンジョセ，1998年1月）において、Urich Benzlerは、動画シーケンスのために1/4ペル精度の動きベクトルの使用と、MPEG4動画符号化基準のさらに進んだ補間フィルタの使用を提案した。しかしながら、Benzlerは、1/4ペルの動きベクトルの発見にGirodのファストサーチ技法を用いた。Benzlerは、異なる補間フィルタを考慮しなかったが、第1段階で複合フィルタを使用して第2段階でより簡単なフィルタを使用することを提案し、一度に1つのマクロブロックを補間した。この方法は、多量のキャッシュメモリは必要としないが、複雑であり、全ての動きベクトルを1つのマクロブロック内の全ての可能なモード（例えば、16×16，4-8×8，16-4×4等）につき1/4ペル精度で計算して最適のモードを決定するために、計算量が多大である。BenzlerはMAD判定基準を使用して、全シーケンスに対し1/4ペル精度に固定した最適動きベクトルを発見した。従って、最適動き精度を選択する方法は示さなかった。最後に、Benzlerは、1/2及び1/4画素精度のベクトルを符号化するために使用可能な可変長符号（“VLC”）テーブルを用いて動きベクトルを符号化した。

【0011】

【発明が解決しようとする課題】上述の参考文献は、最適化したレート歪み基準を用いて動きベクトルを推定しておらず、かような基準の凸状特性を利用して計算の複雑さを軽減していない。さらに、これらの参考文献は、動きベクトルと精度を符号化する有効な戦略を用いていない。

【0012】

【課題を解決するための手段】本発明の1つの実施形態

は、高画素精度（“分数”又は“部分画素”精度とも称される）の動きベクトルを僅かな計算量の増加で計算することにより従来技術の問題を解決する。

【0013】本発明の戦略を用いると、動画符号化器はかなりの圧縮利得（例えば、動き精度の古典的な選択に比較して30%に達するビットレートの削減）を同等の計算レベルで達成できることが、実験により明らかになっている。動き精度に適応して計算し選択するので、本発明は適応動き精度（“AMA”）方式として記述できる。

【0014】本発明の好ましい実施形態の1つは、部分画素（サブピクセル）空間におけるファストサーチ戦略を用いて最適な動きベクトルをスマートにサーチする。この技法は、1マクロブロックに対する最適な動きベクトルを発見することにより、動き補償動画符号化時の動きベクトルを推定する。第1ステップは、 $V_1$ に中心をもつ予め定められた正方形範囲の部分画素解像度のグリッド内で、第1セットの動きベクトル候補をサーチして最適な動きベクトル $V_2$ を見つけ出す。次に、 $V_2$ に中心をもつ予め定められた正方形範囲の部分画素解像度をもつグリッド内で、第2セットの動きベクトル候補を探索して最適な動きベクトル $V_3$ を見つけ出す。その後、 $V_3$ に中心をもつ予め定められた正方形範囲の部分画素解像度をもつグリッド内で、第3セットの動きベクトル候補を探索してマクロブロックの最適動きベクトルを見つけ出す。

【0015】本発明の別の好ましい実施形態において、高精度の動きベクトル推定技法は異なる段階において異なる補間フィルタを使用して計算の複雑さを低減することができる。

【0016】本発明の別の好ましい実施形態は、レート歪み（RD）に関して最適のベクトルと精度を選択する。この実施形態は、異なる動き精度に従い対応するレート歪み判定基準を用いて最適な動きベクトルと最適な動き精度の両方を決定する。

【0017】本発明のさらに別の好ましい実施形態は、有効な可変長符号化（VLC）法により、動きベクトルと精度を符号化する。この技法は、異なる符号化単位で異なる解釈が可能な、関連する動きベクトル精度に従ったVLCテーブルを使用する。

【0018】本発明の上述及び他の目的、特徴及び利点は、添付図面を参照し本発明の以下の詳細な説明を読めば容易に理解されよう。

【0019】

【発明の実施の形態】本発明の諸方法は、各画像ブロックにおいて動き精度を変更して説明するが、精度を全シーケンスに対して固定又はフレーム毎に変更する場合にも適用できる。本発明は、又、発明の背景において記述したように、Telenorの動画符号化器（及び特殊にはTelenorの符号化器）を使用するものとして

記述する。Telenorの動画符号化器の用語を用いて説明するが、ここに記述する技術は、任意の他の動き補償動画符号化器にも適用できる。

【0020】動画符号化器の多くは、半画素（又は“ $1/2$ ペル”）精度の動きベクトルと双線形補間を用いる。Telenorの符号化器の第1バージョンは、 $1/2$ ペル動きベクトルと双線形補間を用いる。しかしながら、Telenorの符号化器の最新バージョンは、さらなる圧縮利得を得るために $1/3$ ペルベクトルと立方体形補間機能を内蔵している。特に、1つの与えられたマクロブロックにおいて、Telenorの符号化器は、図2に示す2つのステップで最適な動きベクトルを推定する。先ず、このTelenor符号化器は、最適な整数ペルベクトル $V_1$ （図1）を探索する（ステップ100）。次に、このTelenor符号化器は、 $V_1$ 付近の最適な $1/3$ ペル精度ベクトル $V_{1/3}$ （図1）をサーチする（ステップ102）。この第2ステップは図1のグラフに示すように、 $3 \times 3$ 内挿（補間）参照フレーム内の（各々 $16 \times 16$ の画素列を有する）全部で8個のブロックを調べて最も一致するブロックを見つけ出す。8個のブロックの動きベクトルは、 $V_1$ に中心をもつグリッド中に8個の実点で表示してある。図1において、最も一致するのは、動きベクトル $V_{1/3} = (V_x, V_y) = (1 + 1/3, 1)$ に関連するブロックである。

【0021】本発明の技術によれば、符号化器は、フルサーチ戦略又はファストサーチ戦略のいずれかを用いて任意の動き精度の組（例えば、 $1/2$ 、 $1/3$ 及び $1/6$ ペル精度動きベクトル）間で精度を選択できる。

【0022】（フルサーチAMA方式のサーチ戦略）図3及び図4に示すように、このフルサーチ適応動き精度（AMA）方式サーチ戦略の場合、Telenor符号化器は、図3に示す5個の画素の（上方画素数、下方画素数及び両側の画素数により規定される正方形ブロックとして定義される）“正方形範囲”と $1/6$ 画素解像度のグリッド上の全ての動きベクトル候補をサーチする。図4に示すように、フルサーチAMAの第1ステップ（104）では最適な整数ペルベクトル $V_1$ （図1）をサーチする。フルサーチAMAの第2ステップ（106）において、符号化器は $V_1$ 付近の最適な $1/6$ 画素精度ベクトル $V_{1/6}$ （図3）をサーチする。換言すれば、このフルサーチAMAは、Telenorプロセスの第2ステップを変更して符号化器が速度空間内の他の部分画素位置における動きベクトル候補もサーチできるようにしたものである。この目的は、グリッド内の最適動きベクトル、即ち、現マクロブロックに最も一致するブロック（補間参照フレーム内）を指示するベクトルを発見することである。このフルサーチ戦略は、120個の部分画素候補をサーチするので計算は複雑ではあるが、本発明のこの実施形態の全潜在能力を示している。

【0023】動きベクトルサーチにおける重要な課題

は、所与のマクロブロックに最も符合するブロックを確定する判定測度又は基準の選択である。実際には、殆どの方法は、平均自乗誤差 (MSE) 又は平均絶対差分 (MAD) の何れかの判定基準を用いている。2つのブロック間のMSEは、2つのブロックの画素値を引き算して、画素値差を自乗し、平均値を取る。2つのブロック間のMAD差分は、自乗計算の代わりに画素値差の絶対値を計算することを除き同様な歪み測度である。2つの画像ブロックが互いに類似であれば、MSE及びMADの値は小さい。しかしながら、画像ブロックが類似でなければ、これらの値は大きい。従って、代表的な動画符号化器は、最小のMSE又は最小のMADの何れかをもちたす動きベクトルを選択することにより、マクロブロックに最も符合するマクロブロックを見つけ出す。言い換えれば、最適動きベクトルに関連するブロックは、MSE又はMADにおいて所与のマクロブロックに最も近似のブロックである。

【0024】残念ながら、MSE及びMADの歪み測度は、ベクトルを実際に符号化するビットのコストを考慮していない。例えば、1つの任意動きベクトルはMSEを最小にできるが、しかし、ビットで符号化するコストが非常に高く、符号化の観点から最適の選択でない場合があり得る。

【0025】この問題を処理するために、Telenorが記述しているような最新の符号化器は、“歪み+L×ビット”形のレート歪み (RD) 判定基準を使用して最適な動きベクトルを選択する。“歪み”値は、代表的にはMSE又はMADであり、“L”は圧縮レベル (即ち、量子化ステップサイズ) に依存する定数であり、“ビット”は動きベクトルの符号化に要するビット数である。一般に、このタイプのどのRD判定基準も本発明に使用できる。しかしながら、本発明の場合、“ビット”は、ベクトルの符号化に要するビットとそのベクトル精度の符号化に要するビットを含んでいる。事実、幾つかの候補が、幾つかの精度モードをもつために、若干数の“ビット”値をもち得る。例えば、位置 (1/2, -1/2) に在る候補は、1/2又は1/6画素精度をもつと考えられる。

【0026】(ファストサーチAMA方式サーチ戦略) 図5及び図6に示すように、ファストサーチ適応動き精度 (AMA) 方式サーチ戦略の場合、符号化器は動きベクトル候補の小さいセットのみを調べる。ファストサーチAMAの第1ステップ (108) では、符号化器は、1/2画素解像度のグリッドにおいてV<sub>1</sub>に中心をもち一辺が1の正方形 (正方形範囲1内) の8個の動きベクトル候補をチェックする。最小のRDコストを有する候補 (即ち、8個の前ベクトルとV<sub>1</sub>の中で最適のもの) を表現するためにV<sub>2</sub>を設定する (110)。次に、符号化器は、1/6画素解像度のグリッド上でV<sub>2</sub>に中心をもち一辺が1の正方形内の8個の動きベクトル位置を

チェックする (112)。V<sub>2</sub>が最小RDコストを有している場合 (114)、符号化器は探索を中止し、V<sub>2</sub>をブロックの動きベクトルとして選択する。そうでなければ、8個の前ベクトルの中で最適のものを表現するためにV<sub>3</sub>を設定する (116)。符号化器は、次に1/6画素解像度のグリッドにおいてV<sub>3</sub>に中心をもつ正方形範囲1内の新しい動きベクトル候補をサーチする (118)。このグリッド内の幾つかの候補は既にチェック済みであり飛ばすことができることに注意すべきである。この最終ステップでは最小のRDコストをもつ候補をブロックの動きベクトルとして選択する (120)。

【0027】実験データによれば、この簡単なファストサーチ戦略は、平均で (Telenor探索戦略より10個多い) 部分画素空間における代表的な約18の位置のRDコストをチェックするので、従って、全体としての計算の複雑さは適度の増加に留まる。

【0028】図8乃至図18に関連して以下に説明する実験結果は、このAMA方式のファストサーチバージョンを用いることによる圧縮性能の損失が実際に皆無であることを示している。その理由は、このファストサーチAMA探索戦略が、RDコストが高レベルから低レベルにスマートに移行するパスを創設することによって“歪み+L×ビット”曲線の凸度 (“歪み”は凸性であることは知られている) を活用しているからである。

【0029】本発明の別の実施態様は、ステップ108-120のうちの1つ又はそれ以上を変更する。これらの実施形態も効率的であり、部分画素速度空間においてチェックする動きベクトル候補数をさらに削減している。

【0030】図7の例では、1/3ペル精度の候補をチェックする。この実施例において、ステップ112は3つの可能なシナリオの1つで置き換えられる。まず、ステップ110からの最適動きベクトル候補がV<sub>1</sub> (“整数ペルベクトル”) の中心にあれば (130)、符号化器は中心ベクトルと次に最低のRDコストをもつ1/2ペル位置の間にある1/3ペル精度の3候補をチェックする (132)。次に、ステップ110からの最適動きベクトル候補がコーナーのベクトルであれば (134)、符号化器はかようなコーナーベクトルに最も近い1/3ペル精度の4ベクトル候補をチェックする (136)。第3に、ステップ110からの最適動きベクトル候補が2つのコーナーベクトルの間にあれば (138)、符号化器は2つのコーナーベクトルの中でRDコストが低い方を決定し、ステップ110からの最適動きベクトル候補とかようなコーナーとの間の直線に最も近い1/3ペル精度の4ベクトル候補をチェックする (140)。V<sub>2</sub>が中心に無く又コーナーベクトルでも無い場合は2つのコーナー間にある筈であり、このプロセスを実行する際にステップ138は不必要になることに注意すべきである。符号化器を1/3画素精度

の動きベクトルの探索に設定する場合、図7は、ステップ114への続行ではなく終了になるように変更可能である。

【0031】(計算量とメモリの削減)ステップ108では1/2画素精度の動きベクトル候補のみをチェックするために、ハードウェアとソフトウェアの実装に要する計算量とメモリは大幅に削減される。特に、このファストサーチのスマートな実装において、参照フレームは2×2だけ補間して1/2ペルベクトル候補に対するRDコストを得る。ハードウェア又はソフトウェア符号化器用のファスト(又はキャッシュ)メモリは、参照フレームを3×3だけ補間するために要するTelenorの方法と比較してかなりの量削減される。Telenorの符号化器と比較し、キャッシュメモリは9/4の削減又は2.25倍の削減になる。若干の追加補間はブロック単位で後で行うことができる。

【0032】さらに、ステップ108における補間はRDコスト関数値を下げる方向にサーチを方向づけるように用いるので、これらの補間には複合フィルタを必要としない。従って、計算量はステップ108用に簡単な双線形フィルタを用いることにより節減できる。

【0033】又、マクロブロックモード(例えば、16×16、4個の8×8等)の選択のような他の主要な符号化に関する決定は、そのような決定は高精度を用いることによる有意な利益がないので、1/2ペルベクトルを用いて行うことができる。次に、符号化器は残りのステップにおいてチェックする若干の追加ベクトル候補に対して必要な部分画素値を補間するためにより複雑な立方体(3次)フィルタを使用することができる。マクロ

ブロックモードは既に選択済みであるので、これらの最後の補間は選択モードに対し実施する必要がある。

【0034】複数フィルタを使用することにより、常時立方体補間を用いるTelenorの方法と比較し、Sparc Ultra 10型ワークステーションでの稼動時間で20%を超える計算量の削減が得られた。さらに、ファストメモリの必要性は、略半分に減少した。又、圧縮性能に関しては損失は殆どなかった。このファストサーチの1実施形態での比較において、Benzler技法は、Telenor符号化器において画素当たり約70の補間を必要とするのに対し、本発明は画素当たり約7つの補間を要するのみである。

【0035】(動きベクトルと精度のビット符号化)最適な動きベクトルと精度が決定されると、符号化器は動きベクトルと精度値の両方をビットで符号化する。1つの方法は、所与の精度(例えば、半画素精度)で動きベクトルを符号化し、次に、そのベクトルをより高い動き精度に詳細化するために幾つかのエクストラビットを追加する。これは、B. Giroudが提案した戦略であるが、但し、レート歪みの点では部分最適化である。

【0036】本発明の1つの好ましい実施形態では、1つのマクロブロックに対する動きベクトルの精度を表1に示したような簡単な符号を用いて最初に符号化する。符号長{1, 2, 2}の任意の他のテーブルを用いることも可能である。ビットレートは、代表的なDPCM法を用いてさらに減少させることができる。

【0037】

【表1】

符号	動き精度
1	1/2ペル
01	1/3ペル
11	1/6ペル

表1. 所与のマクロブロックに対する精度モードを示す可変長(VLC)符号表

【0038】次に、各々の精度空間におけるベクトル値を符号化する。これらのビットはH26Lコーディックにおいて使用されるような単一VLCテーブルの項目より得ることができる。重要なアイデアは、これらのビットはマクロブロックの動き精度によって異なって解釈されることである。例えば、動き精度が1/3であり、異なる動きベクトルのX成分に対する符号ビットが000011である場合、ベクトルのX成分は $V_x = 2/3$ である。精度が1/2であれば、ベクトルのX成分は $V_x = 1$ に相当する。

【0039】1/2及び1/4画素精度ベクトルの符号化に使用できる可変長符号(VLC)テーブルにより動きベクトルを符号化するBenzlerの方法と比較して、本発明の方法は、任意の動き精度のベクトルを符号化するのに用いられ、テーブルは、各フレームとマクロ

ブロックにおいて異なって解釈できる。さらに、本発明の全体的方法は、任意の動き精度に適用でき、互いに倍数であったり $1/n$ ( $n$ は整数)型である必要はない。所与の部分画素空間における増分数は単純に計数し、テーブルの関連項目中のビットを符号として使用する。復号器の観点からは、動き精度を復号すれば動きベクトルもまた容易に復号することができる。その後、前フレーム中の関連ブロックを代表的な4タップの立方体補間器を用いて再構築する。各動き精度毎に異なる4タップフィルタが存在する。AMA方式は、予測ブロックの再構築に要する動作数は同じで、動き精度とは無関係なので、復号の複雑さを増大させることはない。

【0040】(実験結果)図8乃至図18は、表2に記載の種々の動画シーケンス、解像度及びフレームレートでAMAを使用又は使用せずにTelenorの符号化



コーディックを試験した結果を示している。これらの図面は、各ケース毎にレート歪み（“RD”）を作図したものである。“Anchor”曲線は、最適化H. 263+（図8と図9のみ）からのRD点を示している。“Telenor 1/2+b”曲線は1/2ペルベクトルと双線形補間によるTelenor（古典的な例）を示している。“Telenor 1/3”曲線は、現Telenorの提案（“Telenor符号化器”）を示している。“Telenor+AMA+c”曲線は、本発明のフルサーチ戦略によるTelenor符号化器を示している。図15乃至図17に示す“Telenor+F

SAMA+c”曲線は、本発明のファストサーチ戦略によるTelenor符号化器を示している。（他に規定がない限り、AMAのフルサーチバージョンは実験に使用した符号化器の戦略であった。）全ての試験結果は、符号化器と復号器においてクロスチェックを行った。これらの結果は、AMAを実施する場合、信号対雑音（ノイズ）比のピーク（“PSNR”）における利得は、H 26Lを超え1dBの高さであり、古典的な例よりもさらに高い。

【0041】

【表2】

ビデオシーケンス	図#	解像度	フレームレート
コンテナ	図 8	QCIF	10
ニュース	図 9	QCIF	10
自動車	図10	QCIF	10
	図11	SIF	15
庭園	図12	QCIF	15
Tempe te	図13	SIF	15
	図14	QCIF	15
ゆれるバリ	図15	QCIF	10

表2. 実験の説明

【0042】動画シーケンスは、“ゆれるバリ”を除き、動画符号化共同体により共通に使用されている。後者は、よく知られているシーケンス“バリ”を、XとY成分が[-1, 1]の範囲で任意の値をとる動きベクトルだけ移動させることにより得られた合成動画である。この合成動画は、代表的な動画電話シーンにおいて手持ちカメラにより生じた小さな動きをシミュレートする。

【0043】（動き精度適応（AMA）方式のフルサーチ及びファストサーチの比較）図16及び図17に示した実験結果は、AMAに関するファストサーチ戦略（“Telenor FSAMA+c”）とフルサーチ戦略（Telenor AMA+c）方式の符号化器の使用性能が実際には同じであることを示している。ファストサーチ戦略は部分画素速度空間においてRDコスト曲線の凸性を利用するのでこれは間違いない。言い換えれば、RDコスト曲線の形状はなだらかな凸曲線を辿るので、その最小値は、曲線を下降させる幾つかのスマートなファストサーチスキームを用いて容易に見出される。

【0044】（AMA及び複数参照フレームの結合）図18に示すグラフにおいて、“1r”の標識をもつ曲線は、動き補償のために1枚の参照フレームのみを使用した。従って、これらの曲線は図10に示した曲線と同じである。“5r”の標識をもつ曲線は5枚の参照フレームを使用した。AMAによる利得は複数の参照フレームを使用して得た曲線に付加されることを実験は示している。参照フレーム1枚の場合のAMAによる利得は、緑とピンクの曲線（注：凡例で×と△）を比較することにより測定でき、参照フレーム5枚の場合の利得は青と赤

の曲線（注：凡例で○と●）間で測定できる。

【0045】特に注意すべきことは、本発明はフレームレベルで実施でき、異なるフレームは異なる動き精度を用いることができるが、1フレーム内の全ての動きベクトルには同一精度を用いることである。この実施形態において、動きベクトル精度は、フレーム層において1度だけ信号化されるのが好ましい。実験は、全フレームに対して最適な固定動き精度を用いることによって、マクロブロック適応の場合につき提示したような圧縮利得が生じることを示している。

【0046】他のフレームベースの実施形態において、符号化器は、異なるベクトル精度でフレーム全体に動き補償を行い、その後、RD判定基準に従って最適な精度を選択することができる。この方法は、パイプライン方式のワンパス符号化器には適しないが、ソフトウェアベースの符号化器又はより複雑な符号化には適用できる。他のフレームベースの実施形態において、符号化器は所与のフレームに対する最適精度を予測するために以前の統計値及び／又は公式を用いることができる（例えば、Ribasの論文に記述されている公式又はそのバージョンを使用することができる）。この方法は、性能利得は予測に用いる公式の精度に依存するが、ワンパス符号化器には適している。

【0047】上述の明細書に用いた用語と表現は説明のためのものであり制限するものではなく、かような用語と表現を用いるに当たって表示及び記述の特徴又はその部分の同等の用語及び表現を排除する意図は全くなく、本発明の範囲は特許請求範囲の請求項によってのみ規定

され限定されることを確認する。

【図面の簡単な説明】

【図1】速度空間における完全ベル及び1/3ベルの位置の例を示す図である。

【図2】最適動きベクトルを推定する従来技術による方法を説明するためのフローチャート図である。

【図3】部分画素速度空間におけるフルサーチ時の動きベクトル候補位置の例を示す図である。

【図4】本発明によるフルサーチ方式の最適動きベクトル推定方法の好ましい実施態様を説明するためのフローチャート図である。

【図5】部分画素速度空間におけるファストサーチ時の動きベクトル候補位置の例を示す図である。

【図6】本発明によるファストサーチ方式の最適動きベクトル推定方法の好ましい実施形態を説明するためのフローチャート図である。

【図7】図6のステップ114の別の好ましい実施形態を説明するためのフローチャート図である。

【図8】動画シーケンス“コンテナ”において每秒10フレームのフレームレートと解像度QCIFで適応動き精度(AMA)を使用、または使用しないでTelenor符号化器の性能試験をした結果を示すグラフ図である。

【図9】動画シーケンス“ニュース”において每秒10フレームのフレームレートと解像度QCIFで適応動き精度(AMA)を使用または使用しないでTelenor符号化器の性能試験をした結果を示すグラフ図である。

【図10】動画シーケンス“自動車”において每秒10フレームのフレームレートと解像度QCIFで適応動き精度(AMA)を使用または使用しないでTelenor符号化器の性能試験をした結果を示すグラフ図である。

【図11】動画“庭園”において毎秒15フレームのフレームレートと解像度SIFで適応動き精度(AMA)を使用または使用しないでTelenor符号化器

の性能試験をした結果を示すグラフ図である。

【図12】動画シーケンス“庭園”において毎秒15フレームのフレームレートと解像度QCIFで適応動き精度(AMA)を使用または使用しないでTelenor符号化器の性能試験をした結果を示すグラフ図である。

【図13】動画シーケンス“Tempete”において毎秒15フレームのフレームレートと解像度SIFで適応動き精度(AMA)を使用または使用しないでTelenor符号化器の性能試験をした結果を示すグラフ図である。

【図14】動画シーケンス“Tempete”において毎秒15フレームのフレームレートと解像度QCIFで適応動き精度(AMA)を使用または使用しないでTelenor符号化器の性能試験をした結果を示すグラフ図である。

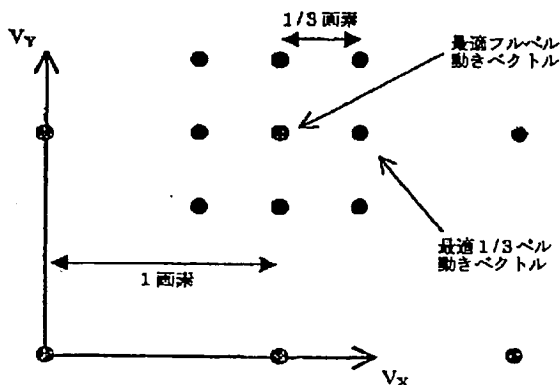
【図15】動画シーケンス“ゆれるバリ”において毎秒15フレームのフレームレートと解像度QCIFで適応動き精度(AMA)を使用または使用しないでTelenor符号化器の性能試験をした結果を示すグラフ図である。

【図16】動画シーケンス“自動車”において、毎秒10フレームのフレームレートと解像度QCIFでファストサーチ戦略(“Telenor FSAMA+c”)及びフルサーチ戦略(“Telenor AMA+c”)の性能試験の結果を示すグラフ図である。

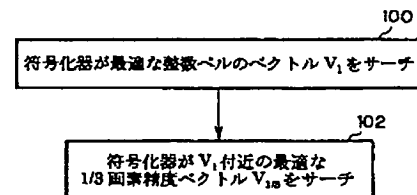
【図17】動画シーケンス“コンテナ”において毎秒10フレームのフレームレートと解像度QCIFでファストサーチ戦略(“Telenor FSAMA+c”)及びフルサーチ戦略(“Telenor AMA+c”)の性能試験の結果を示すグラフ図である。

【図18】動画シーケンス“自動車”において解像度QCIFと毎秒10フレームのフレームレートで動き補償のために1枚の参照フレームのみを使用し試験した場合と動き補償のために複数枚のフレームを使用し試験した場合とを比較する性能試験結果を示すグラフ図である。

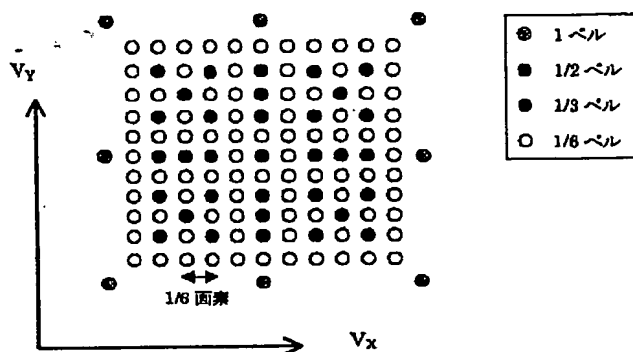
【図1】



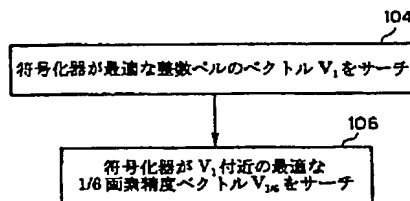
【図2】



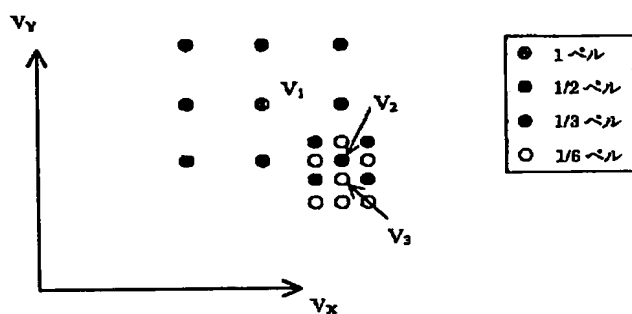
【図3】



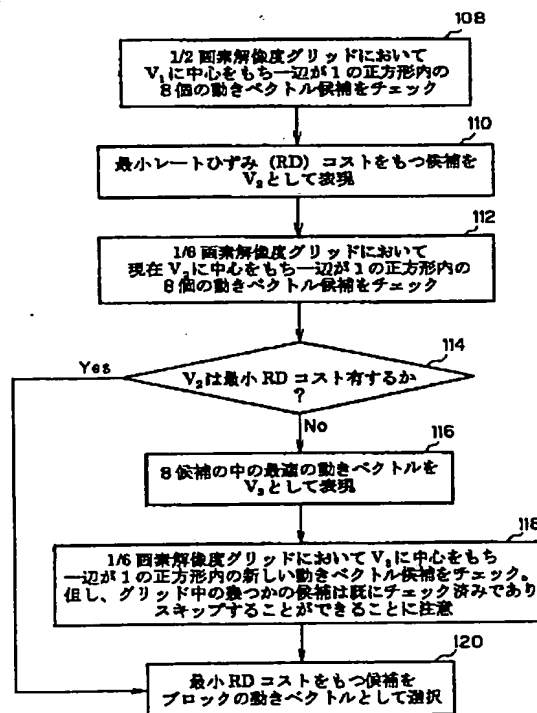
【図4】



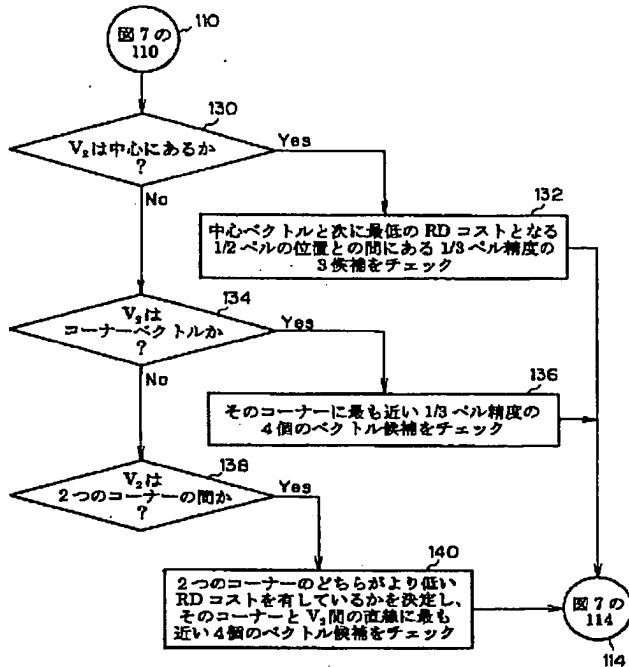
【図5】



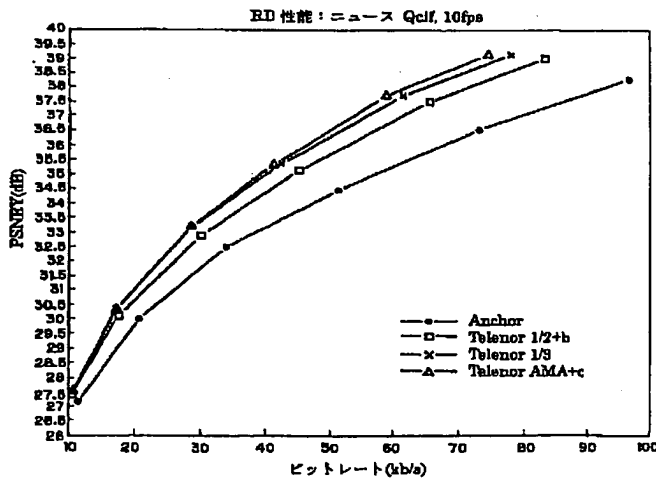
【図6】



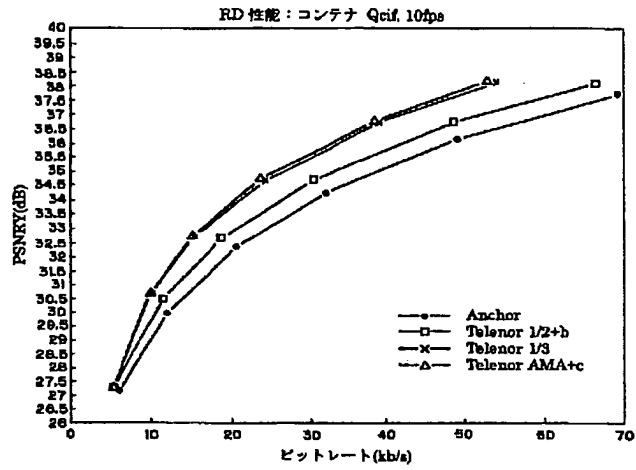
【図7】



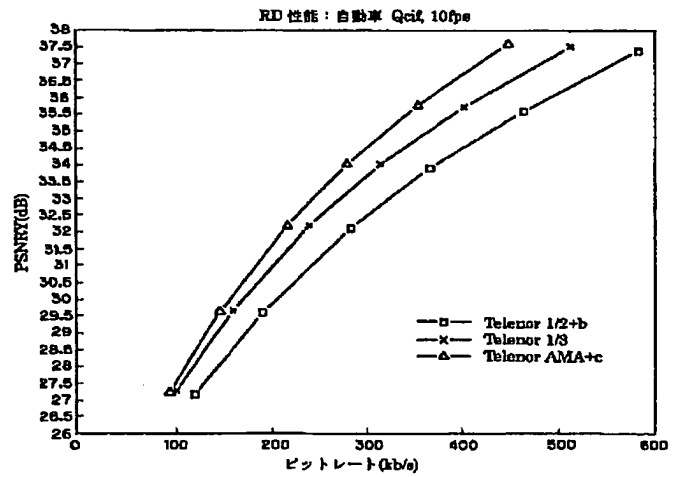
【図9】



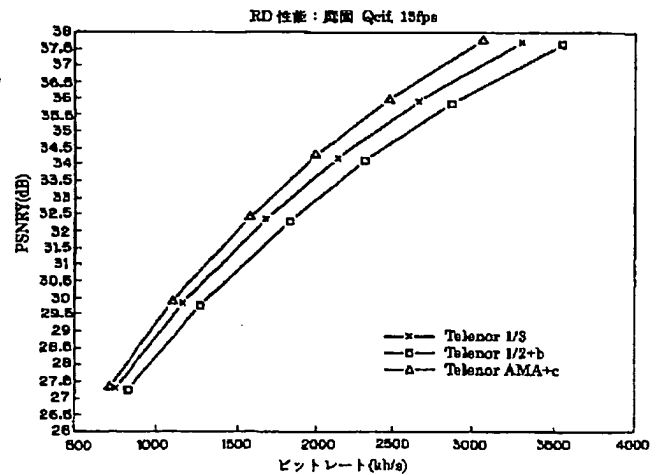
【図8】



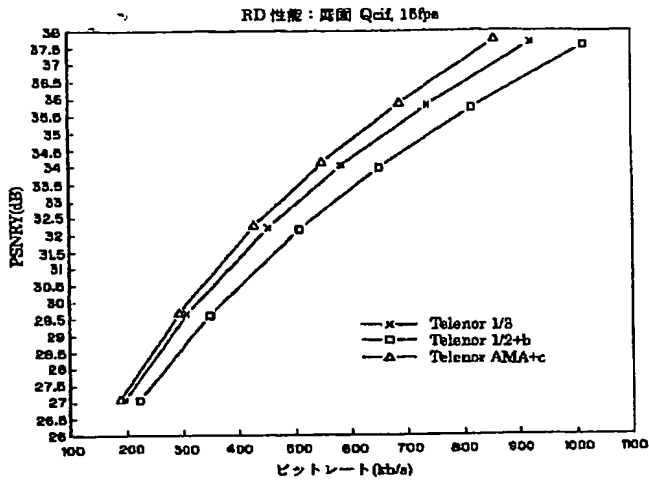
【図10】



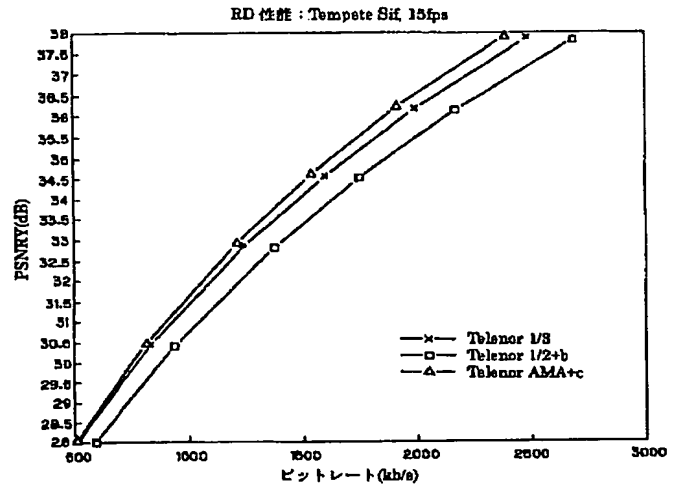
【図11】



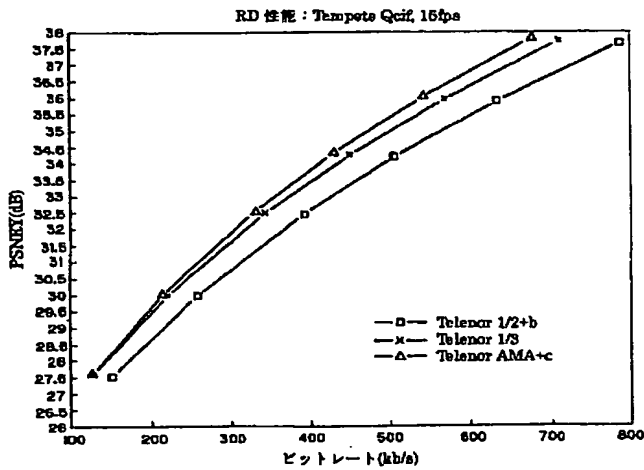
【図12】



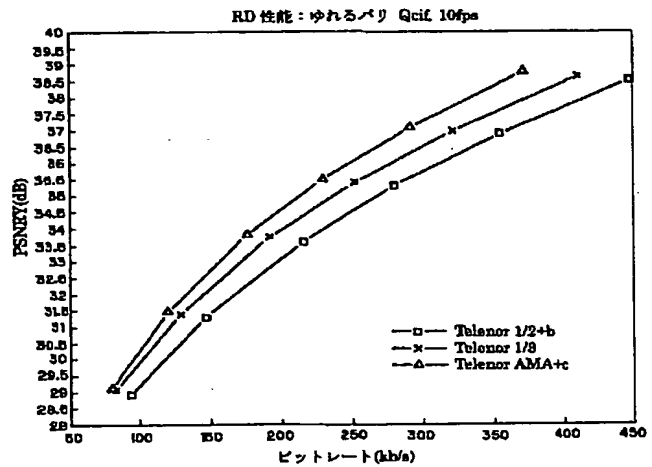
【図13】



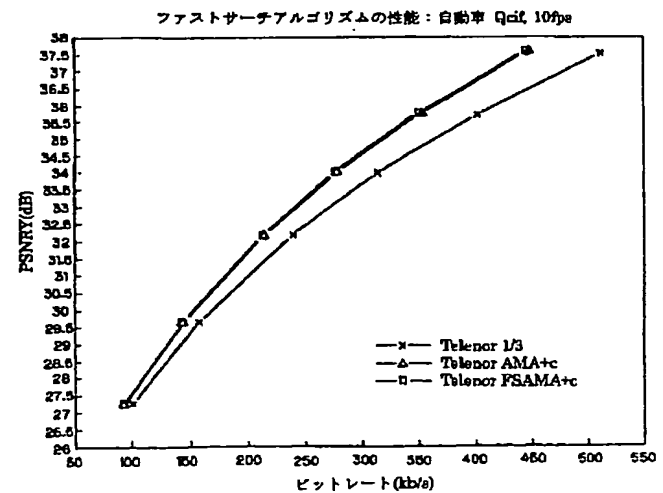
【図14】



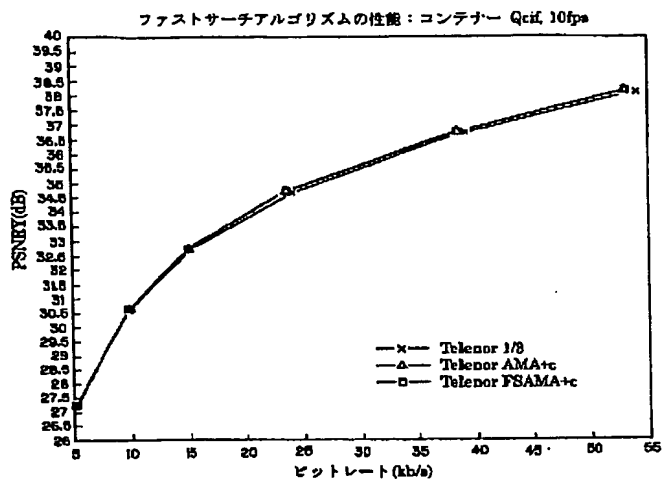
【図15】



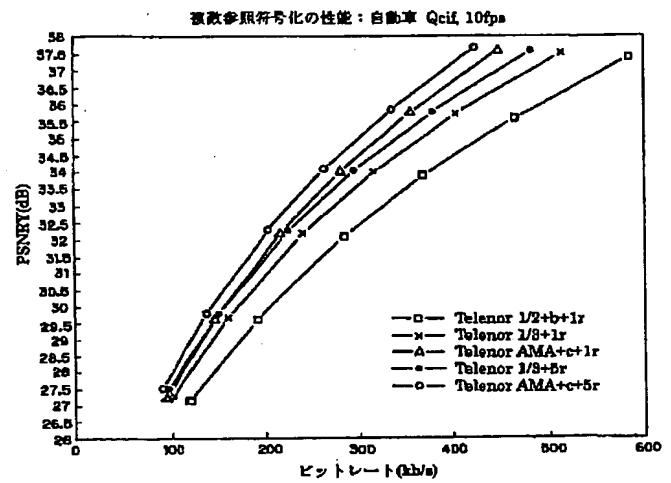
【図16】



【図17】



【図18】



## 【外国語明細書】

## 1. Title of Invention

METHODS FOR MOTION ESTIMATION WITH  
ADAPTIVE MOTION ACCURACY

## 2. Claims

1. A fast-search adaptive motion accuracy search method for estimating motion vectors in motion-compensated video coding by finding a best motion vector for a macroblock, said method comprising the steps of:

- (a) searching a first set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_1$  to find a best motion vector  $V_2$ ;
- (b) searching a second set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_2$  to find a best motion vector  $V_3$ ; and
- (c) searching a third set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_3$  to find said best motion vector of said macroblock.

2. The method of claim 1, said step of searching a first set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_1$  to find a best motion vector  $V_2$  further comprising the step of searching a first set of eight motion vector candidates in a grid of 1/2-pixel resolution of square radius 1 centered on  $V_1$  to find a best motion vector  $V_2$ .

3. The method of claim 1, said step of searching a second set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_2$  to find a best motion vector  $V_3$  further comprising the step of searching a second set of eight motion vector candidates in a grid of 1/6-pixel resolution of square radius 1 centered on  $V_2$  to find a best motion vector  $V_3$ .

4. The method of claim 1 further comprising the steps of using  $V_2$  as the motion vector for the block if  $V_2$  has the smallest rate-distortion cost and skipping step (c) of claim 1.

5. The method of claim 1, said step of searching a third set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_3$  to find said best motion vector of said macroblock further comprising the step of searching a third set of eight motion vector candidates in a grid of 1/6-pixel resolution of square radius 1 centered on  $V_3$  to find said best motion vector of said macroblock.

6. The method of claim 1, said step of searching a third set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_3$  to find said best motion vector of said macroblock further comprising the step of skipping motion vector candidates of said third set of motion vector candidates that have already been tested.

7. The method of claim 1 further wherein said step of searching said first set of motion vector candidates further comprises the step of searching said first set of motion vector candidates using a first filter to do a first interpolation, said step of searching said second set of motion vector candidates further comprises the step of searching said second set of motion vector candidates using a second filter to do a second interpolation, and said step of searching said third set of motion vector candidates further comprises the step of searching said third set of motion vector candidates using a third filter to do a third interpolation.

8. The method of claim 1, said step of searching a second set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_2$  to find a best motion vector  $V_3$  further comprising the steps of:

- (a) searching three candidates of 1/3-pel accuracy  $V_2$  and a 1/2-pel location with the next lowest RD cost if  $V_2$  is at the center;
- (b) searching four vector candidates of 1/3-pel accuracy that are closest to  $V_2$  if  $V_2$  is a corner vector; and



- (c) determining which of two corners has lower RD cost and searching four vector candidates of 1/3-pel accuracy that are closest to a line between said corner with lower RD cost, if  $V_2$  is between two corners vectors.

9. An adaptive motion accuracy search method for estimating motion vectors in motion-compensated video coding by finding a best motion vector for a macroblock, said method comprising the steps of:

- (a) searching a first set of motion vector candidates in a grid centered on  $V_1$  to find a best motion vector  $V_2$  using a first filter to do a first interpolation;
- (b) searching a second set of motion vector candidates in a grid centered on  $V_2$  to find a best motion vector  $V_3$  using a second filter to do a second interpolation; and
- (c) searching a third set of motion vector candidates in a grid centered on  $V_3$  to find said best motion vector of said macroblock using a third filter to do a third interpolation.

10. The method of claim 9 wherein said step of searching using a first filter to do a first interpolation further comprises using a simple filter to do a coarse interpolation.

11. The method of claim 9 wherein said step of searching using a first filter to do a first interpolation further comprises using a simple filter to do a coarse interpolation and said step of searching using a second filter to do a second interpolation further comprises using a complex filter to do a fine interpolation.

12. The method of claim 11 wherein said step of searching using a third filter to do a third interpolation further comprises using a complex filter to do a fine interpolation.

13. The method of claim 9 wherein said step of searching using a first filter to do a first interpolation further comprises using a bilinear filter to interpolate the reference frame by 2x2.

14. The method of claim 9 wherein said step of searching using a first filter to do a first interpolation further comprises using a bilinear filter to interpolate the reference frame by 2x2 and said step of searching using a second filter to do a second interpolation further comprises using a cubic filter to do a fine interpolation.

15. The method of claim 14 wherein said step of searching using a third filter to do a third interpolation further comprises using a cubic filter to do a fine interpolation.

16. An adaptive motion accuracy search method for estimating motion vectors in motion-compensated video coding by finding a best motion vector for a macroblock, said method comprising the steps of:

- (a) searching at a first motion accuracy for a first best motion vector of said macroblock;
- (b) encoding said first best motion vector and said first motion accuracy;
- (c) searching for at least one second best motion vector of said macroblock at an at least one second motion accuracy;
- (d) encoding said at least one second best motion vector and said at least one second motion accuracy; and
- (e) selecting the best motion vector of said first and at least one best motion vectors using rate-distortion criteria.

17. The method of claim 16 wherein said step of selecting the best motion vector using rate-distortion criteria further comprises the step of said rate-distortion criteria adapting according to the different motion accuracies to determine both the best motion vectors and the best motion accuracies.

18. The method of claim 16, said step of searching for at least one second best motion vector at an at least one second motion accuracy further comprising the step of searching for at least one second best motion vector of said macroblock at an at least one second motion accuracy that is finer than said first motion accuracy.

19. The method of claim 16 wherein said step of selecting the best motion vector using rate-distortion criteria further comprises the step of using rate-distortion criteria of the type "distortion + L\*Bits" to select the best motion vector.

20. An adaptive motion accuracy search method for estimating motion vectors in motion-compensated video coding by finding a best motion vector for a macroblock, said method comprising the steps of:

- (a) searching at a motion accuracy for a best motion vector of said macroblock;
- (b) encoding said motion accuracy using a code from a VLC table that is interpreted differently at different coding units according to the associated motion vector accuracy; and
- (c) encoding said best motion vector in the respective accuracy space.

21. A system for estimating motion vectors in motion-compensated video coding by finding a best motion vector for a macroblock, said system comprising:

- (a) a first encoder for searching a first set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_1$  to find a best motion vector  $V_2$ ;
- (b) a second encoder for searching a second set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_2$  to find a best motion vector  $V_3$ ; and
- (c) a third encoder for searching a third set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_3$  to find said best motion vector of said macroblock.

22. The system of claim 21 wherein said first, second, and third encoders are a single encoder.

### 3. Detailed Description of Invention

#### BACKGROUND OF THE INVENTION

The present invention relates generally to a method of compressing or coding digital video with bits and, specifically, to an effective method for estimating and encoding motion vectors in motion-compensated video coding.

In classical motion estimation the current frame to be encoded is decomposed into image blocks of the same size, typically blocks of 16x16 pixels, called "macroblocks." For each current macroblock, the encoder searches for the block in a previously encoded frame (the "reference frame") that best matches the current macroblock. The coordinate shift between a current macroblock and its best match in the reference frame is represented by a two-dimensional vector (the "motion vector") of the macroblock. Each component of the motion vector is measured in pixel units.

For example, if the best match for a current macroblock happens to be at the same location, as is the typical case in stationary background, the motion vector for the current macroblock is (0,0). If the best match is found two pixels to the right and

- three pixels up from the coordinates of the current macroblock, the motion vector is (2,3). Such motion vectors are said to have integer pixel (or "integer-pel" or "full-pel") accuracy, since their horizontal X and vertical Y components are integer pixel values. In FIG. 1, the vector  $V_1 = (1,1)$  represents the full-pel motion vector for a given current macroblock.

Moving objects in a video scene do not move in integer pixel increments from frame to frame. True motion can take any real value along the X and Y directions. Consequently, a better match for a current macroblock can often be found by interpolating the previous frame by a factor  $N \times N$  and then searching for the best match in the interpolated frame. The motion vectors can then take values in increments of  $1/N$  pixel along X and Y and are said to have  $1/N$  pixel (or " $1/N$ -pel") accuracy.

In "Response to Call for Proposals for H.26L," ITU-Telecommunications Standardization Sector, Q.15/SG16, doc. Q15-F-11, Seoul, Nov. 98, and "Enhancement of the Telenor proposal for H.26L," ITU-Telecommunications Standardization Sector, Q.15/SG16, doc. Q15-G-25, Monterey, Feb. 99, Gisle Bjontegaard proposed using  $1/3$ -pel accurate motion vectors and cubic-like interpolation for the H26L video coding standard (the "Telenor encoder"). To do this, the Telenor encoder interpolates or "up-samples" the reference frame by  $3 \times 3$  using a cubic-like interpolation filter. This interpolated version requires nine times more memory than the reference frame. At a given macroblock, the Telenor encoder estimates the best motion vector in two steps: the encoder first searches for the best integer-pel vector and then the Telenor encoder searches for the best  $1/3$ -pixel accurate vector  $V_{1/3}$  near  $V_1$ . Using FIG. 1 as an example, a total of eight blocks (of  $16 \times 16$  pixels) in the  $3 \times 3$  interpolated reference frame are checked to find the best match which, as shown is the block associated to the motion vector  $V_{1/3} = (VX, VY) = (1+1/3, 1)$ . The Telenor encoder has several problems. First, it uses a sub-optimal fast-search strategy and a complex cubic filter (at all stages) to compute the  $1/3$ -pel accurate motion vectors. As a result, the computed motion vectors are not optimal and the memory and computation requirements are very expensive. Further, the Telenor encoder uses an accuracy of the effective rate-distortion criteria that is fixed at  $1/3$ -pixel and, therefore, does not adapt to select better motion accuracies.

Similarly, the Telenor encoder variable-length code ("VLC") table has an accuracy fixed at 1/3-pixel and, therefore, is not adapted and interpreted differently for different accuracies.

Most known video compression methods estimate and encode motion vectors with 1/2-pixel accuracy, because early studies suggested that higher or adaptive motion accuracies would increase computational complexity without providing additional compression gains. These early studies, however, did not estimate the motion vectors using optimized rate-distortion criteria, did not exploit the convexity properties of such criteria to reduce computational complexity, and did not use effective strategies to encode the motion vectors and their accuracies.

One such early study was Bernd Girod's "Motion-Compensating Prediction with Fractional-Pel Accuracy," IEEE Transactions on Communications, Vol. 41, No. 4, pp. 604-612, April 1993 (the "Girod work"). The Girod work is the first fundamental analysis on the benefits of using sub-pixel motion accuracy for video coding. Girod used a simple, hierarchical strategy to search for the best motion vector in sub-pixel space. He also used simple mean absolute difference ("MAD") criteria to select the best motion vector for a given accuracy. The best accuracy was selected using a formula that is not useful in practice since it is based on idealized assumptions, is very complex, and restricts all motion vectors to have the same accuracy within a frame. Finally, Girod focused only on prediction error energy and did not address how to use bits to encode the motion vectors.

Another early study was Smita Gupta's and Allen Gersho's "On Fractional Pixel Motion Estimation," Proc. SPIE VCIP, Vol. 2094, pp. 408-419, Cambridge, November 1993 (the "Gupta work"). The Gupta work presented a method for computing, selecting, and encoding motion vectors with sub-pixel accuracy for video compression. The Gupta work disclosed a formula based on mean squared error ("MSE") and bilinear interpolation, used this formula to find an ideal motion vector, and then quantized such vector to the desired motion accuracy. The best motion vector for a given accuracy was found using the sub-optimal MSE criteria and the best accuracy was selected using the largest decrease in difference energy per distortion bit, which is a greedy (sub-optimal)

criteria. A given motion vector was coded by first encoding that vector with 1/2-pel accuracy and then encoding the higher accuracy with refinement bits. Course-to-fine coding tends to require significant bit overhead.

In "On the Optimal Motion Vector Accuracy for Block-Based Motion-Compensated Video Coders," Proc. IST/SPIE Digital Video Compression: Algorithms and Technologies, pp. 302-314, San Jose, February 1996 (the "Ribas work"), Jordi Ribas-Corbera and David L. Neuhoff, modeled the effect of motion accuracy on bit rate and proposed several methods to estimate the optimal accuracies that minimize bit rate. The Ribas work set forth a full-search approach for computing motion vectors for a given accuracy and considered only bilinear interpolation. The best motion vector was found by minimizing MSE and the best accuracy was selected using some formulas derived from a rate-distortion optimization. The motion vectors and accuracies were encoded with frame-adaptive entropy coders, which are complex to implement in real-time applications.

In "Proposal for a new core experiment on prediction enhancement at higher bitrates," ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, MPEG 97/1827, Sevilla, Feb. 1997 and "Performance Evaluation of a Reduced Complexity Implementation for Quarter Pel Motion Compensation," ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, MPEG 97/3146, San Jose, Jan. 1998, Ulrich Benzler proposed using 1/4-pel accurate motion vectors for the video sequence and more advanced interpolation filters for the MPEG4 video coding standard. Benzler, however, used the Girod's fast-search technique to find the 1/4-pel motion vectors. Benzler did consider different interpolation filters, but proposed a complex filter at the first stage and a simpler filter at the second stage and interpolated one macroblock at a time. This approach does not require much cache memory, but it is computationally expensive because of its complexity and because all motion vectors are computed with 1/4-pel accuracy for all the possible modes in a macroblock (e.g., 16x16, four-8x8, sixteen-4x4, etc.) and then the best mode is determined. Benzler used the MAD criteria to find the best motion vector which was fixed to 1/4-pel accuracy for the whole sequence, and hence he did not address how to select the best motion accuracy. Finally,

Benzler encoded the motion vectors with a variable-length code ("VLC") table that could be used for encoding 1/2 and 1/4 pixel accurate vectors.

The references discussed above do not estimate the motion vectors using optimized rate-distortion criteria and do not exploit the convexity properties of such criteria to reduce computational complexity. Further, these references do not use effective strategies to encode motion vectors and their accuracies.

#### BRIEF SUMMARY OF THE INVENTION

One preferred embodiment of the present invention addresses the problems of the prior art by computing motion vectors of high pixel accuracy (also denoted as "fractional" or "sub-pixel" accuracy) with a minor increase in computation.

Experiments have demonstrated that, by using the search strategy of the present invention, a video encoder can achieve significant compression gains (e.g., up to thirty percent in bit rate savings over the classical choices of motion accuracy) using similar levels of computation. Since the motion accuracies are adaptively computed and selected, the present invention may be described as adaptive motion accuracy ("AMA").

One preferred embodiment of the present invention uses fast-search strategies in sub-pixel space that smartly searches for the best motion vectors. This technique estimates motion vectors in motion-compensated video coding by finding a best motion vector for a macroblock. The first step is searching a first set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_1$  to find a best motion vector  $V_2$ . Next, a second set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_2$  is searched to find a best motion vector  $V_3$ . Then, a third set of motion vector candidates in a grid of sub-pixel resolution of a predetermined square radius centered on  $V_3$  is searched to find the best motion vector of the macroblock.

In an alternate preferred embodiment the present invention a technique for estimating high-accurate motion vectors may use different interpolation filters at different stages in order to reduce computational complexity.

Another alternate preferred embodiment of the present invention selects the best vectors and accuracies in a rate-distortion ("RD") sense. This embodiment uses



rate-distortion criteria that adapts according to the different motion accuracies to determine both the best motion vectors and the best motion accuracies.

Still further, another alternate preferred embodiment of the present invention encodes the motion vector and accuracies with an effective VLC approach. This technique uses a VLC table that is interpreted differently at different coding units, according to the associated motion vector accuracy.

The foregoing and other objectives, features, and advantages of the invention will be more readily understood upon consideration of the following detailed description of the invention, taken in conjunction with the accompanying drawings.

#### DETAILED DESCRIPTION OF THE INVENTION

The methods of the present invention are described herein in terms of the motion accuracy being modified at each image block. These methods, however, may be applied when the accuracy is fixed for the whole sequence or modified on a frame-by-frame basis. The present invention is also described as using Telenor's video encoders (and particularly the Telenor encoder) as described in the Background of the Invention. Although described in terms of Telenor's video encoders, the techniques described herein are applicable to any other motion-compensated video coder.

Most video coders use motion vectors with half pixel (or "1/2-pel") accuracy and bilinear interpolation. The first version of Telenor's encoder also used 1/2-pel motion vectors and bilinear interpolation. The latest version of Telenor's encoder, however, incorporated 1/3-pel vectors and cubic-like interpolation because of the additional compression gains. Specifically, at a given macroblock, Telenor's estimates the best motion vector in two steps shown in FIG. 2. First, the Telenor encoder searches for the best integer-pel vector  $V_1$  (FIG. 1) 100. Second, the Telenor encoder searches for the best 1/3-pixel accurate vector  $V_{1/3}$  (FIG. 1) near  $V_1$  102. This second step is shown graphically in FIG. 1 where a total of eight blocks (each having an array of 16x16 pixels) in the 3x3 interpolated reference frame are checked to find the best match. The motion vectors for these eight blocks are represented by the eight solid dots in the grid centered on  $V_1$ . In FIG. 1 the best match is the block associated to the motion vector  $V_{1/3} = (V_x, V_y) = (1+1/3, 1)$ .

The technology of the present invention allows the encoder to choose between any set of motion accuracies (for example, 1/2, 1/3, and 1/6-pel accurate motion vectors) using either a full search strategy or a fast search strategy.

### Full-Search AMA Search Strategy

As shown in FIGS. 3 and 4, in the full-search adaptive motion accuracy ("AMA") search strategy the encoder searches all the motion vector candidates in a grid of 1/6-pixel resolution and a "square radius" (defined herein as a square block defined by a number of pixels up, a number of pixels down, and a number of pixels to both sides) of five pixels as shown in FIG. 3. FIG. 4 shows that the first step of the full-search AMA is to search for the best integer-pel vector  $V_1$  (FIG. 1) 104. In the second step of the full-search AMA, the encoder searches for the best 1/6-pixel accurate vector  $V_{1/6}$  (FIG. 3) near  $V_1$  106. In other words, the full-search AMA modifies the second step of the Telenor's process so that the encoder also searches for motion vector candidates in other sub-pixel locations in the velocity space. The objective is to find the best motion vector in the grid, i.e., the vector that points to the block (in the interpolated reference frame) that best matches the current macroblock. Although the full-search strategy is computationally complex since it searches 120 sub-pixel candidates, it shows the full potential of this preferred method of the present invention.

A critical issue in the motion vector search is the choice of a measure or criterion for establishing which block is the best match for the given macroblock. In practice, most methods use either the mean squared error ("MSE") or mean absolute difference ("MAD") criteria. The MSE between two blocks consists of subtracting the pixel values of the two blocks, squaring the pixel differences, and then taking the average. The MAD difference between two blocks is a similar distortion measure, except that the absolute value of the pixel differences is computed instead of the squares. If two image blocks are similar to each other, the MSE and MAD values will be small. If, however, the image blocks are dissimilar, these values will be large. Hence, typical video coders find the best match for a macroblock by selecting the motion vector that produces either the smallest MSE or the smallest MAD. In other words, the block associated to the best motion vector is the one closest to the given macroblock in an MSE or MAD sense.

Unfortunately, the MSE and MAD distortion measures do not take into account the cost in bits of actually encoding the vector. For example, a given motion

vector may minimize the MSE, but it may be very costly to encode with bits, so it may not be the best choice from an coding standpoint.

To deal with this, advanced encoders such as those described by Telenor use rate-distortion ("RD") criteria of the type "distortion + L\*Bits" to select the best motion vector. The value of "distortion" is typically the MSE or MAD, "L" is a constant that depends on the compression level (i.e., the quantization step size), and "Bits" is the number of bits required to code the motion vector. In general, any RD criteria of this type would work with the present invention. However, in the present invention "Bits" include the bits needed for encoding the vector and those for encoding the accuracy of the vector. In fact, some candidates can have several "Bits" values, because they can have several accuracy modes. For example, the candidate at location  $(1/2, -1/2)$  can be thought of having  $1/2$  or  $1/6$  pixel accuracy.

#### Fast-Search AMA Search Strategy

As shown in FIGS. 5 and 6, in the fast-search adaptive motion accuracy ("AMA") search strategy the encoder checks only a small set of the motion vector candidates. In the first step of the fast-search AMA, the encoder checks the eight motion vector candidates in a grid of  $1/2$ -pixel resolution of square radius 1, which is centered on  $V_1$  108.  $V_2$  is then set to denote the candidate that has the smallest RD cost (i.e., the best of the eight previous vectors and  $V_1$ ) 110. Next, the encoder checks the eight motion vector locations in a grid of  $1/6$ -pixel resolution of square radius 1 that is now centered on  $V_2$  112. If  $V_2$  has the smallest RD cost 114, the encoder stops its search and selects  $V_2$  as the motion vector for the block. Otherwise,  $V_3$  is set to denote the best motion vector of the eight 116. The encoder then searches for a new motion vector candidates in the grid of  $1/6$ -pixel resolution of square radius 1 that is centered on  $V_3$  118. It should be noted that some of the candidates in this grid have already been tested and can be skipped. The candidate with the smallest RD cost in this last step is selected as the motion vector for the block 120.

Experimental data has shown that, on average, this simple fast search strategy typically checks the RD cost of about eighteen locations in sub-pixel space (ten more than

Telenor's search strategy), and hence the overall computational complexity is only moderately increased.

The experimental data discussed below in connection with FIGS. 8-18 show that there is practically no loss in compression performance from using this fast-search version of AMA. This is because the fast-search AMA search strategy exploits the convexity of the "distortion + L\*Bits" curve (c.f., "distortion" is known to be convex), by creating a path that smartly follows the RD cost from higher to lower levels.

Alternate embodiments of the invention replace one or more of the steps 108-120. These embodiments have also been effective and have further reduced the number of motion vector candidates to check in the sub-pixel velocity space.

FIG. 7, for example, checks candidates of 1/3-pel accuracy. In this embodiment step 112 is replaced by one of three possible scenarios. First, if the best motion vector candidate from step 110 is at the center of  $V_1$  (the "integer-pel vector") 130, then the encoder checks three candidates of 1/3-pel accuracy between the center vector and the 1/2-pel location with the next lowest RD cost 132. Second, if the best motion vector candidate from step 110 is a corner vector 134, then, the encoder checks the four vector candidates of 1/3-pel accuracy that are closest to such corner 136. Third, if the best motion vector candidate from step 110 is between two corners 138, then, the encoder determines which of these two corners has lower RD cost and checks the four vector candidates of 1/3-pel accuracy that are closest to the line between such corner and the best candidate from step 110 140. It should be noted that in implementing this process step 138 may be unnecessary because if  $V_2$  is neither at the center or a corner vector, then it would necessarily be between two corners. If the encoder is set to find motion vectors with 1/3-pixel accuracy, FIG. 7 could be modified to end rather than continuing with step 114.

#### Computation And Memory Savings

Because step 108 checks only motion vector candidates of 1/2-pixel accuracy, the computation and memory requirements for the hardware or software implementation are significantly reduced. To be specific, in a smart implementation

embodiment of this fast-search the reference frame is interpolated by  $2 \times 2$  in order to obtain the RD costs for the  $1/2$ -pel vector candidates. A significant amount of fast (or cache) memory for a hardware or software encoder is saved as compared to Telenor's approach that needed to interpolate the reference frame by  $3 \times 3$ . In comparison to the Telenor encoder, this is a cache memory savings of  $9/4$  or a factor of 2.25. The few additional interpolations can be done later on a block-by-block basis.

Additionally, since the interpolations in step 108 are used to direct the search towards the lower values of the RD cost function, a complex filter is not needed for these interpolations. Accordingly, computation power may be saved by using a simple bilinear filter for step 108.

Also, other key coding decisions such as selecting the mode of a macroblock (e.g.,  $16 \times 16$ , four- $8 \times 8$ , etc.) can be done using the  $1/2$ -pel vectors because such decisions do not benefit significantly from using higher accuracies. Then, the encoder can use a more complex cubic filter to interpolate the required sub-pixel values for the few additional vector candidates to check in the remaining steps. Since the macroblock mode has already been chosen, these final interpolations only need to be done for the chosen mode.

Use of multiple-filters obtained computation savings of over twenty percent in running time on a Sparc Ultra 10 Workstation in comparison to Telenor's approach, which uses a cubic interpolation all the time. Additionally, the fast-memory requirements were reduced by nearly half. Also, there was little or no loss in compression performance. Comparing one preferred embodiment of the fast-search, Benzler's technique requires about 70 interpolations per pixel in the Telenor encoder and the present invention requires only about 7 interpolations per pixel.

#### Coding The Motion Vector And Accuracies With Bits

Once the best motion vector and accuracy are determined, the encoder encodes both the motion vector and accuracy values with bits. One approach is to encode the motion vector with a given accuracy (e.g., half-pixel accuracy) and then add some extra

bits for refining the vector to the higher motion accuracy. This is the strategy suggested by B. Girod, but it is sub-optimal in a rate-distortion sense.

In one preferred embodiment of the present invention, the accuracy of the motion vector for a macroblock is first encoded using a simple code such as the one given in Table 1. Any other table with code lengths {1, 2, 2} could be used as well. The bit rate could be further reduced using a typical DPCM approach.

Code	Motion Accuracy
1	1/2-pel
01	1/3-pel
11	1/6-pel

Table 1. VLC table to indicate the accuracy mode for a given macroblock.

Next, the value of the vector/s in the respective accuracy space is encoded. These bits can be obtained from entries of a single VLC table such as the one used in the H26L codec. The key idea is that these bits are interpreted differently depending on the motion accuracy for the macroblock. For example, if the motion accuracy is 1/3 and the code bits for the X component of the difference motion vector are 000011, the X component of the vector is  $V_x = 2/3$ . If the accuracy is 1/2, such code corresponds to  $V_x = 1$ .

Compared to the Benzler method for encoding the motion vectors with a variable length code ("VLC") table that could be used for encoding 1/2 and 1/4 pixel accurate vectors, the method of the present invention can be used for encoding vectors of any motion accuracy and the table can be interpreted differently at each frame and macroblock. Further, the general method of the present invention can be used for any motion accuracy, not necessarily those that are multiples of each other or those that are of the type 1/n (with n an integer). The number of increments in the given sub-pixel space is simply counted and the bits in the associated entry of the table is used as the code.

From the decoder's viewpoint, once the motion accuracy is decoded, the motion vector can also be easily decoded. After that, the associated block in the previous frame is reconstructed using a typical 4-tap cubic interpolator. There is a different 4-tap filter for each motion accuracy.

The AMA does not increase decoding complexity, because the number of operations needed to reconstruct the predicted block are the same, regardless of the motion accuracy.

### Experimental Results

FIGS. 8-18 show test results of the Telenor encoder codec with and without AMA in a variety of video sequences, resolutions, and frame rates, as described in Table 2. These figures show rate-distortion ("RD") plots for each case. The "Anchor" curve shows RD points from optimized H.263+ (FIGS. 8 and 9 only). The "Telenor 1/2+b" curve shows Telenor with 1/2-pel vectors and bilinear interpolation (the "classical case"). The "Telenor 1/3" curve shows the current Telenor proposal (the "Telenor encoder"). The "Telenor+AMA+c" curve shows the Telenor encoder with the full-search strategy of the present invention. The "Telenor+FSAMA+c", as shown in FIGS. 15-17, shows the current Telenor encoder with the fast-search strategy. (Unless otherwise specified, the full-search version of AMA was the encoder strategy used in the experiments.) All of the test results were cross-checked at the encoder and decoder. These results show that with AMA the gains in peak signal-to-noise ratio ("PSNR") can be as high as 1 dB over H26L, and even higher over the classical case.

Video sequence	FIG. #	Resolution	Frame rate
Container	FIG. 8	QCIF	10
News	FIG. 9	QCIF	10
Mobile	FIG. 10	QCIF	10

	FIG. 11	SIF	15
Garden	FIG. 12	QCIF	15
Tempete	FIG. 13	SIF	15
	FIG. 14	QCIF	15
Paris Shaked	FIG. 15	QCIF	10

Table 2. Description of the Experiments

The video sequences are commonly used by the video coding community, except for "Paris Shaked." The latter is a synthetic sequence obtained by shifting the well-known sequence "Paris" by a motion vector whose X and Y components take a random value within  $[-1,1]$ . This synthetic sequence simulates small movements caused by a hand-held camera in a typical video phone scene.

#### Comparison Of Full-Search And Fast-Search AMA

The experimental results shown in FIGS. 16 and 17 demonstrate that the encoder performance with fast-search ("Telenor FSAMA+c") and full-search ("Telenor AMA+c") strategies for AMA is practically the same. This is true because the fast-search strategies exploit the convexity of the RD cost curve in the sub-pixel velocity space. In other words, since the shape of the RD cost follows a smooth convex curve, its minimum should be easy to find with some smart fast-search schemes that descend down the curve.

#### Combining AMA And Multiple Reference Frames

In the plot shown in FIG. 18, the curves labeled "1r" used only one reference frame for the motion compensation, so these curves are the same as those presented in FIG. 10. The curves labeled "5r" used five reference frames.

The experiments show that the gains with AMA add to those obtained using multiple reference frames. The gain from AMA in the one-reference case can be measured by comparing the green and pink curves, and the gain in the five-reference case can be measured between the blue and red curves.



It should be noted that the present invention may be implemented at the frame level so that different frames could use different motion accuracies, but within a frame all motion vectors would use the same accuracy. Preferably in this embodiment the motion vector accuracy would then be signaled only once at the frame layer. Experiments have shown that using the best, fixed motion accuracy for the whole frame should also produce compression gains as those presented here for the macroblock-adaptive case.

In another frame-based embodiment the encoder could do motion compensation on the entire frame with the different vector accuracies and then select the best accuracy according to the RD criteria. This approach is not suitable for pipeline, one-pass encoders, but it could be appropriate for software-based or more complex encoders. Still another frame-based embodiment the encoder could use previous statistics and/or formulas to predict what will be the best accuracy for a given frame (e.g., the formulas in set forth in the Ribas work or a variation thereof can be used). This approach would be well-suited for one-pass encoders, although the performance gains would depend on the precision of the formulas used for the prediction.

The terms and expressions which have been employed in the foregoing specification are used therein as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding equivalents of the features shown and described or portions thereof, it being recognized that the scope of the invention is defined and limited only by the claims that follow.

#### 4. Brief Description of Drawings

FIG. 1 is a diagram of an exemplary full-pel and 1/3-pel locations in velocity space.

FIG. 2 is a flowchart illustrating a prior art method for estimating the best motion vector.

FIG. 3 is a diagram of an exemplary location of motion vector candidates for full-search in sub-pixel velocity space.

FIG. 4 is a flowchart illustrating a full-search preferred embodiment of the method for estimating the best motion vector of the present invention.

FIG. 5 is a diagram of an exemplary location of motion vector candidates for fast-search in sub-pixel velocity space.

FIG. 6 is a flowchart illustrating a fast-search preferred embodiment of the method for estimating the best motion vector of the present invention.

FIG. 7 is a detail flowchart illustrating an alternate preferred embodiment of step 114 of FIG. 6.

FIG. 8 is a graphical representation of experimental performance results of the Telenor encoder with and without AMA in the "Container" video sequence, with QCIF resolution, and at the frame rate of 10 frames per second.

FIG. 9 is a graphical representation of experimental performance results of the Telenor encoder with and without AMA in the "News" video sequence, with QCIF resolution, and at the frame rate of 10 frames per second.

FIG. 10 is a graphical representation of experimental performance results of the Telenor encoder with and without AMA in the "Mobile" video sequence, with QCIF resolution, and at the frame rate of 10 frames per second.

FIG. 11 is a graphical representation of experimental performance results of the Telenor encoder with and without AMA in the "Garden" video sequence, with SIF resolution, and at the frame rate of 15 frames per second.

FIG. 12 is a graphical representation of experimental performance results of the Telenor encoder with and without AMA in the "Garden" video sequence, with QCIF resolution, and at the frame rate of 15 frames per second.

FIG. 13 is a graphical representation of experimental performance results of the Telenor encoder with and without AMA in the "Tempete" video sequence, with SIF resolution, and at the frame rate of 15 frames per second.

FIG. 14 is a graphical representation of experimental performance results of the Telenor encoder with and without AMA in the "Tempete" video sequence, with QCIF resolution, and at the frame rate of 15 frames per second.

FIG. 15 is a graphical representation of experimental performance results

of the Telenor encoder with and without AMA in the "Paris shaken" video sequence, with QCIF resolution, and at the frame rate of 10 frames per second. .

FIG. 16 is a graphical representation of experimental performance results of fast-search ("Telenor FSAMA+c") and full-search ("Telenor AMA+c") strategies in the "Mobile" video sequence, with QCIF resolution, and at the frame rate of 10 frames per second.

FIG. 17 is a graphical representation of experimental performance results of fast-search ("Telenor FSAMA+c") and full-search ("Telenor AMA+c") strategies in the "Container" video sequence, with QCIF resolution, and at the frame rate of 10 frames per second.

FIG. 18 is a graphical representation of experimental performance results of tests using only one reference frame for motion compensation as compared to tests using multiple reference frames for motion compensation the in the "Mobile" video sequence, with QCIF resolution, and at the frame rate of 10 frames per second.

FIG. 1

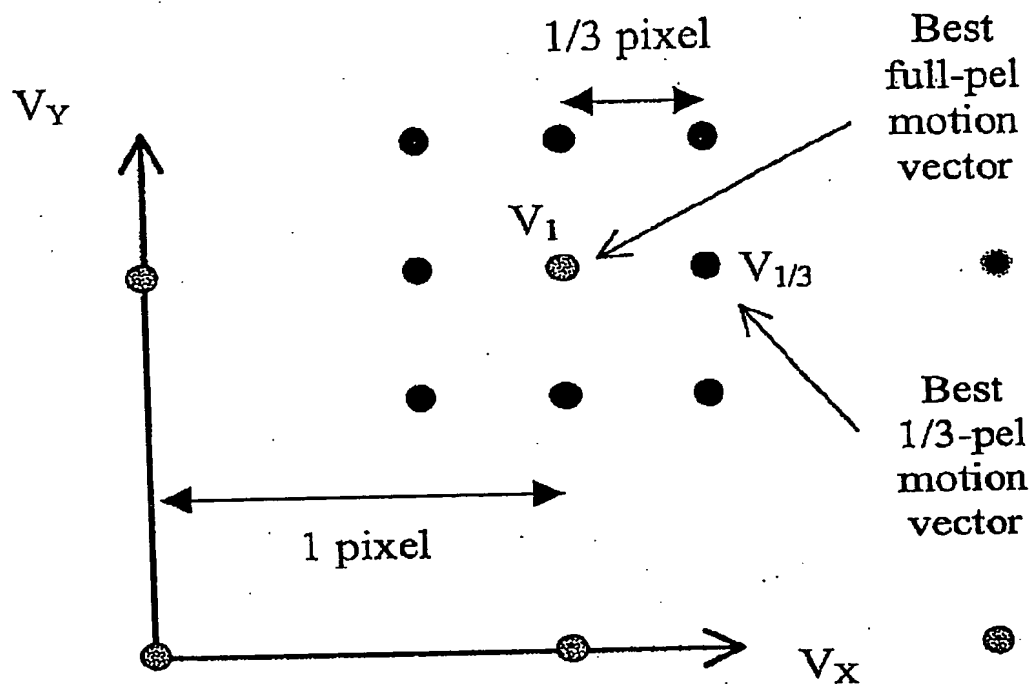


FIG.2

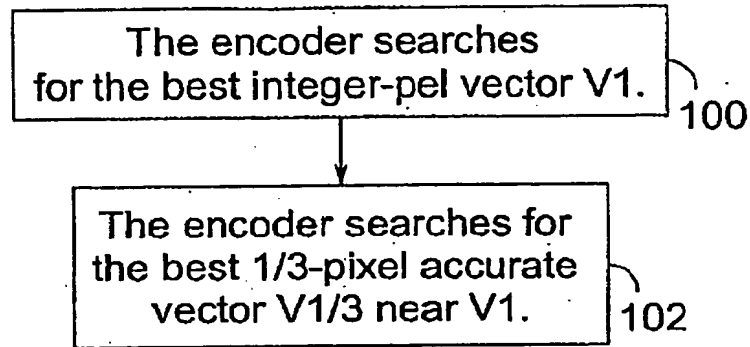


FIG.3

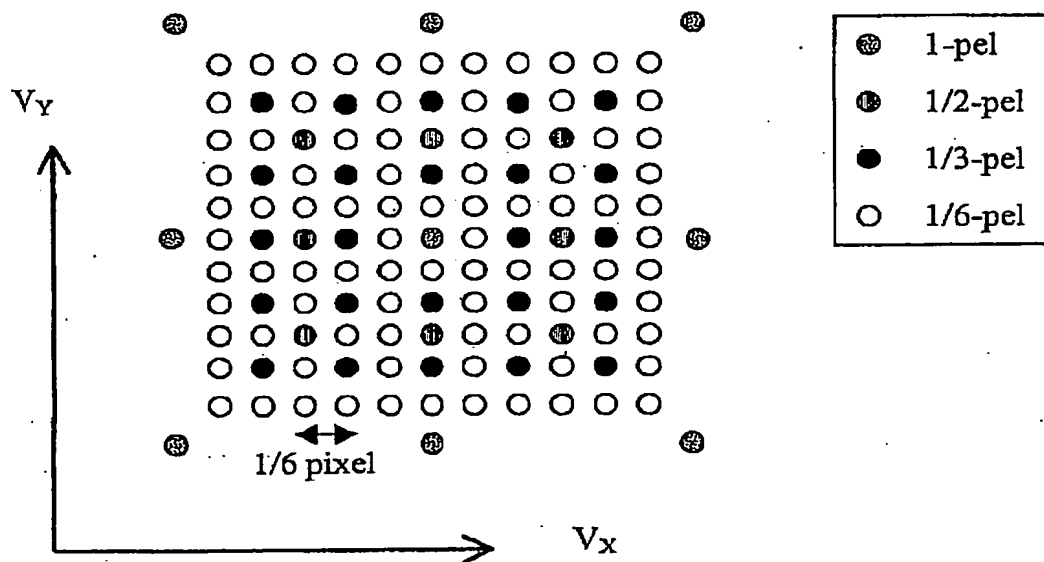


FIG.4

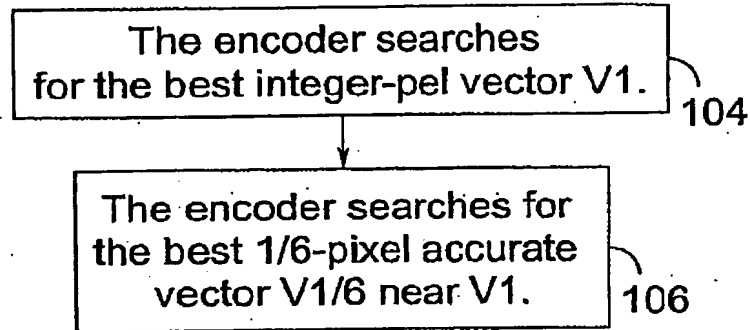


FIG.5

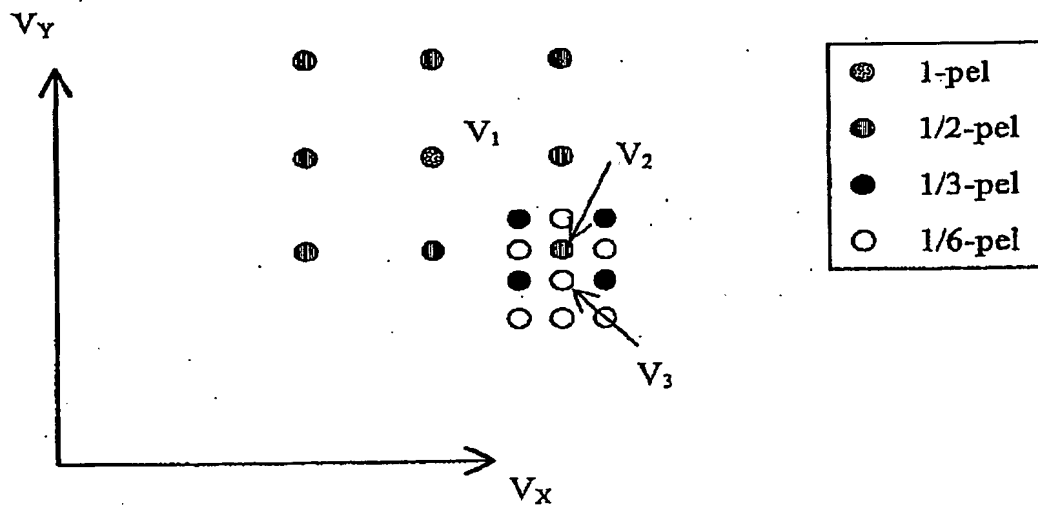


FIG.6

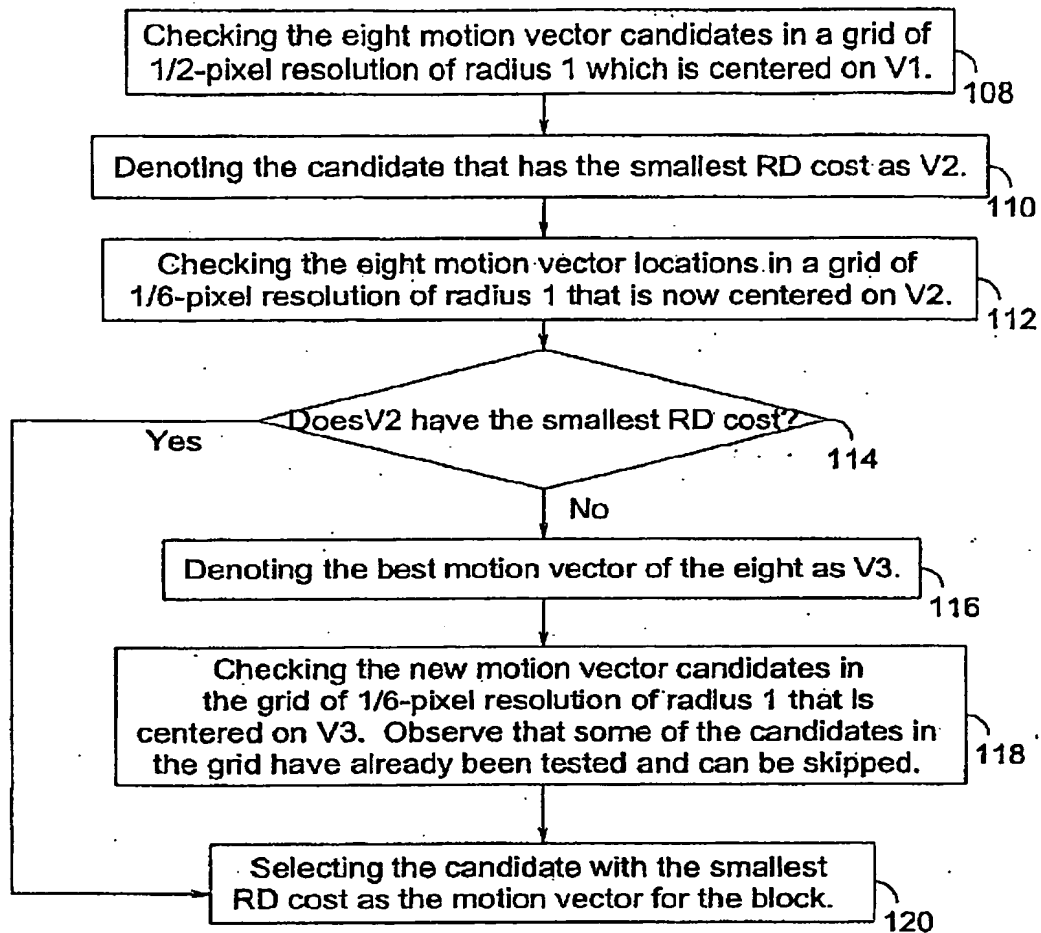


FIG.7

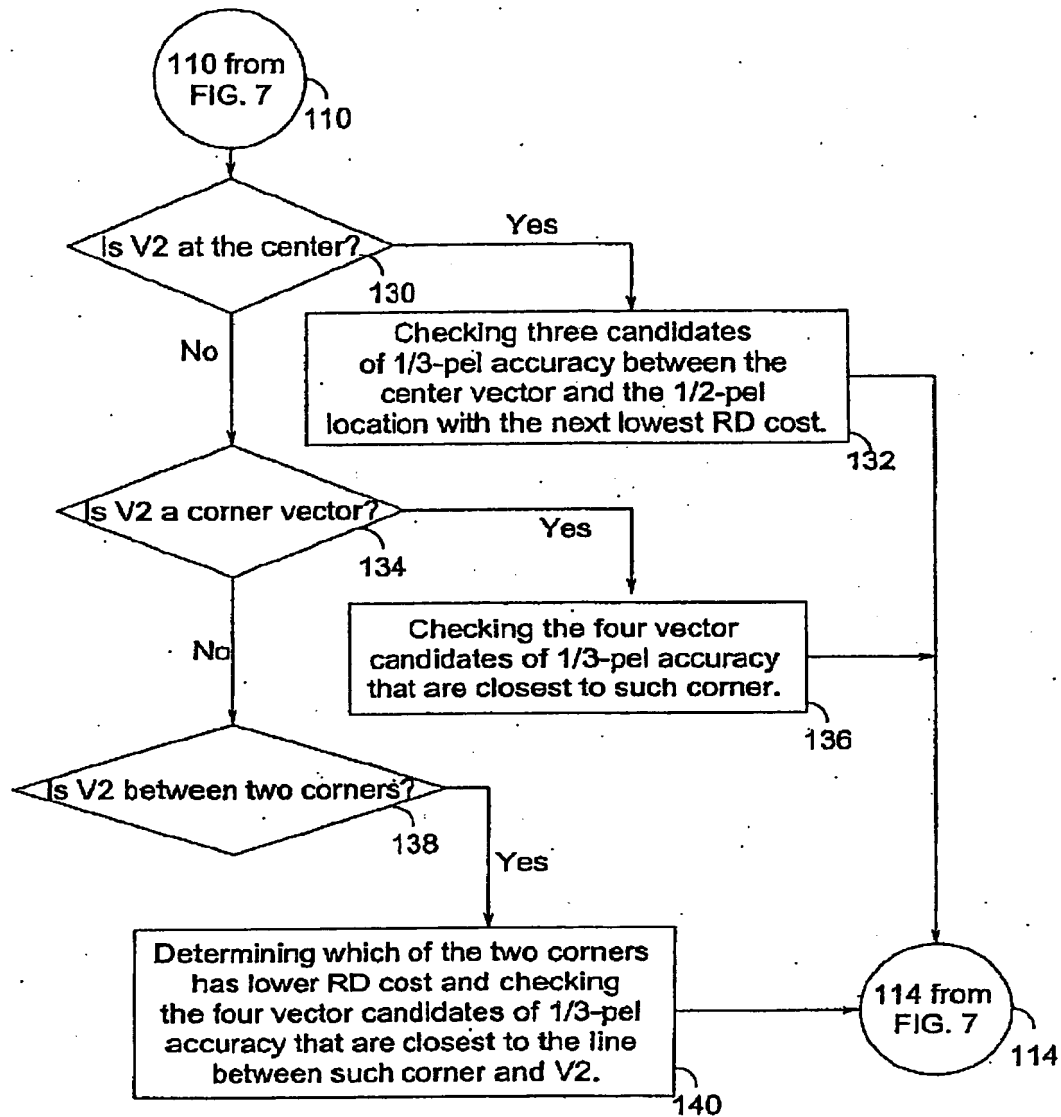




FIG.8

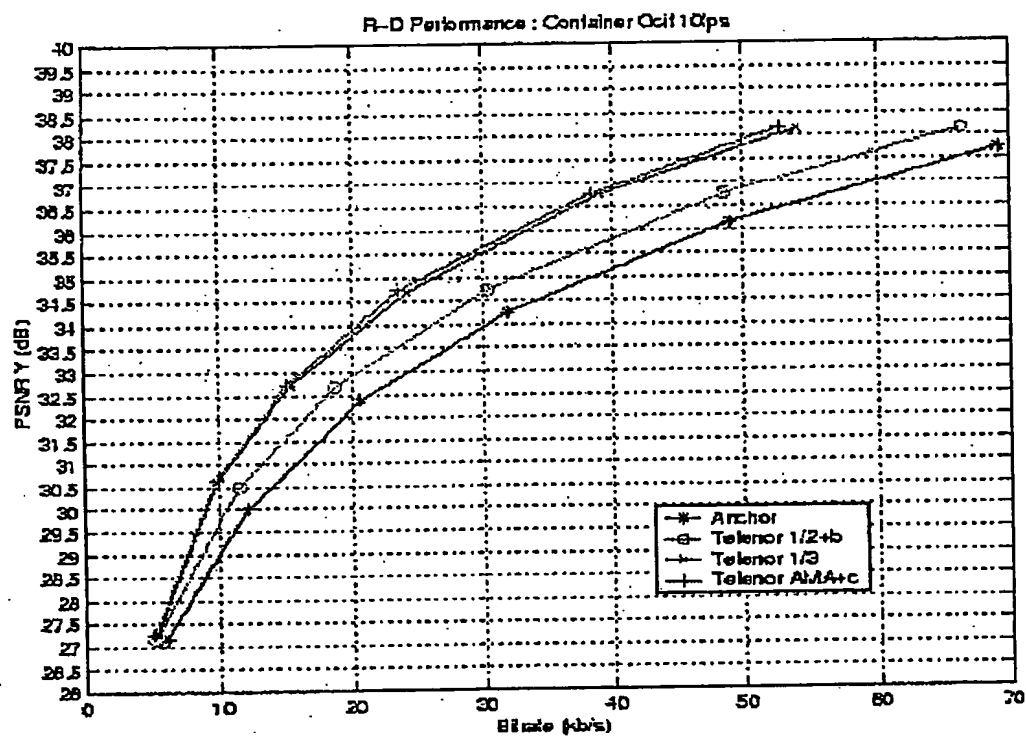


FIG.9

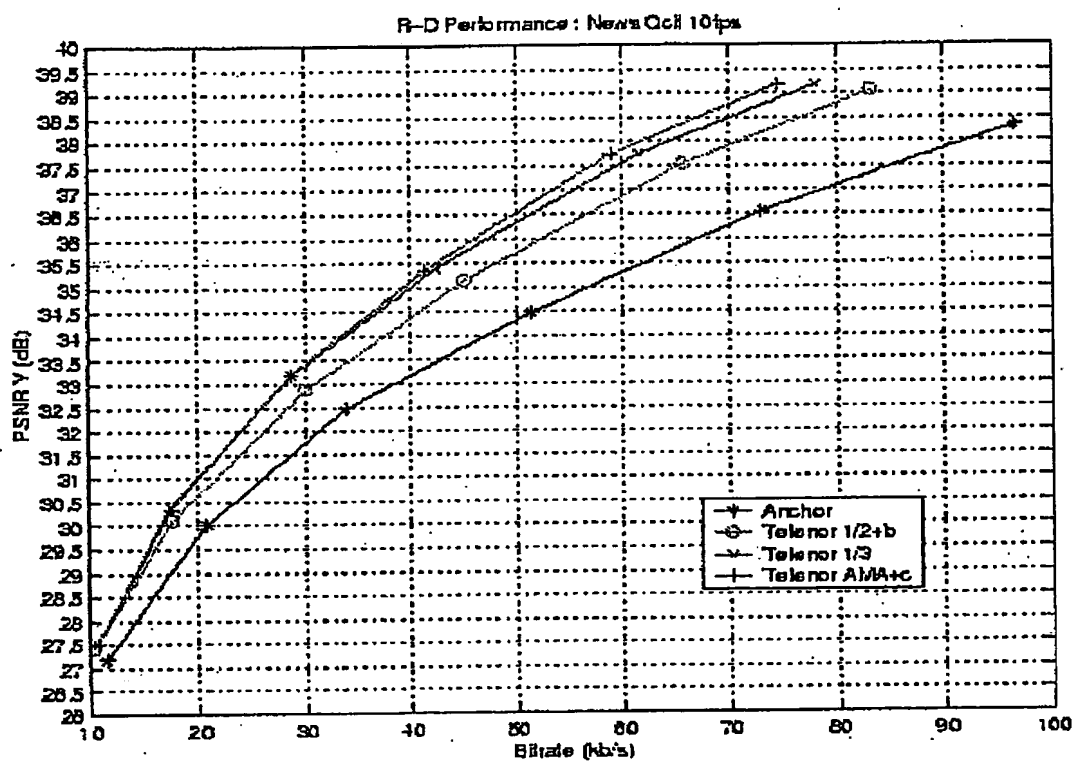


FIG.10

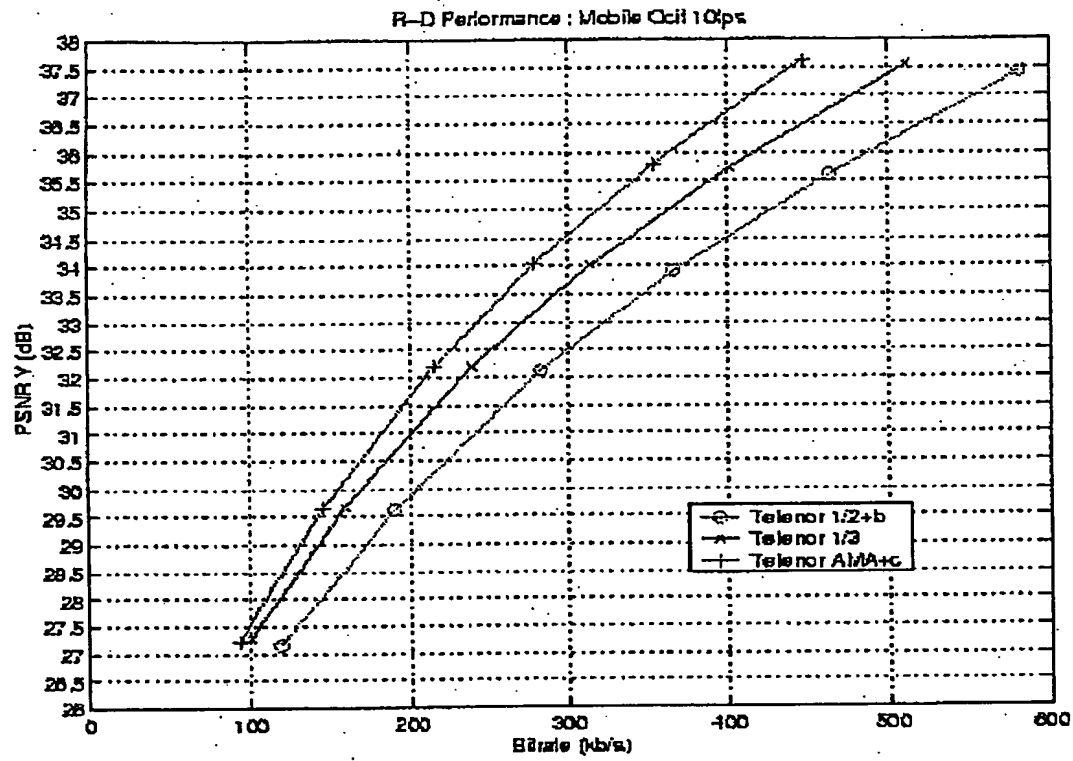


FIG.11

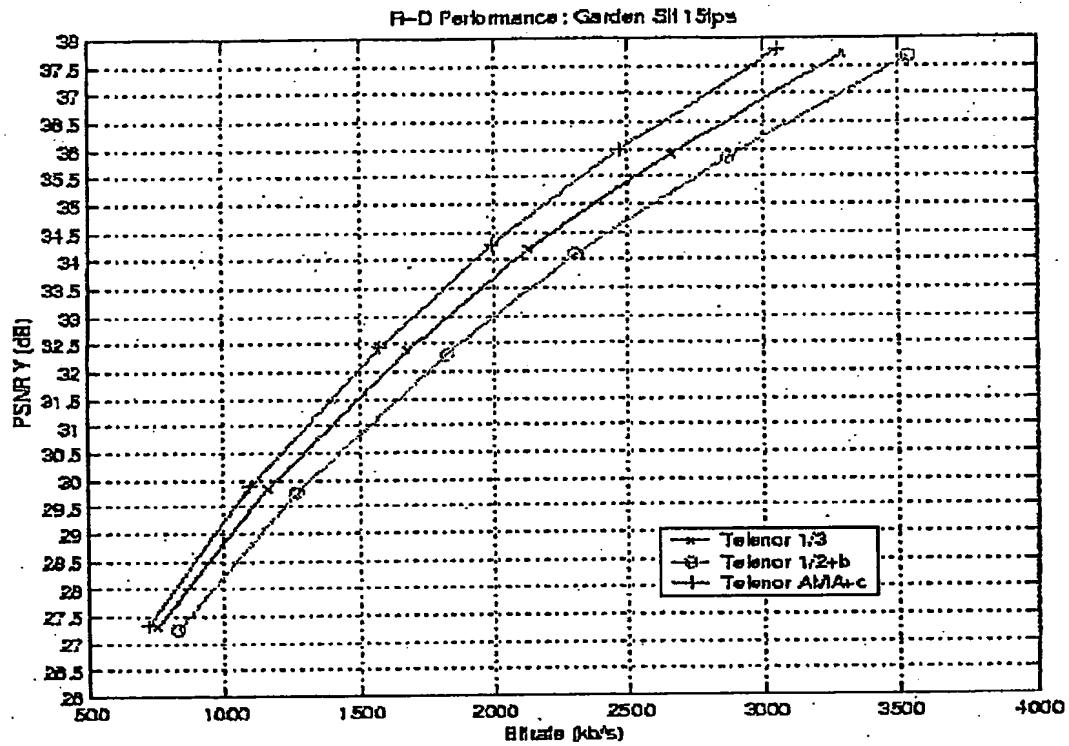


FIG.12

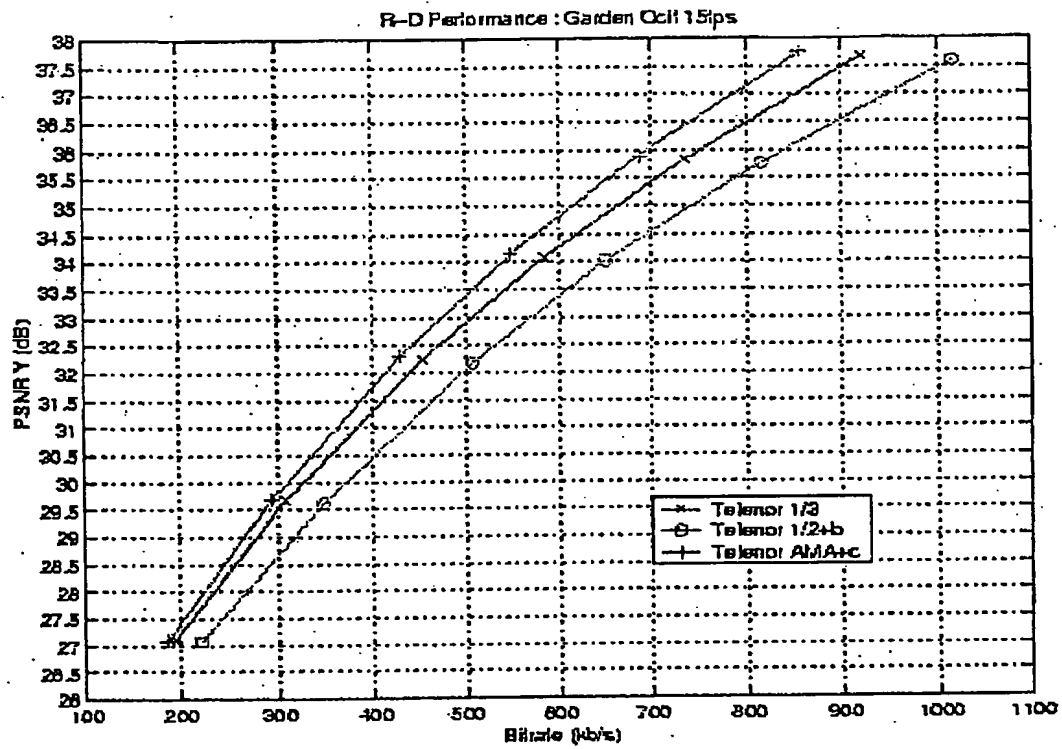


FIG.13

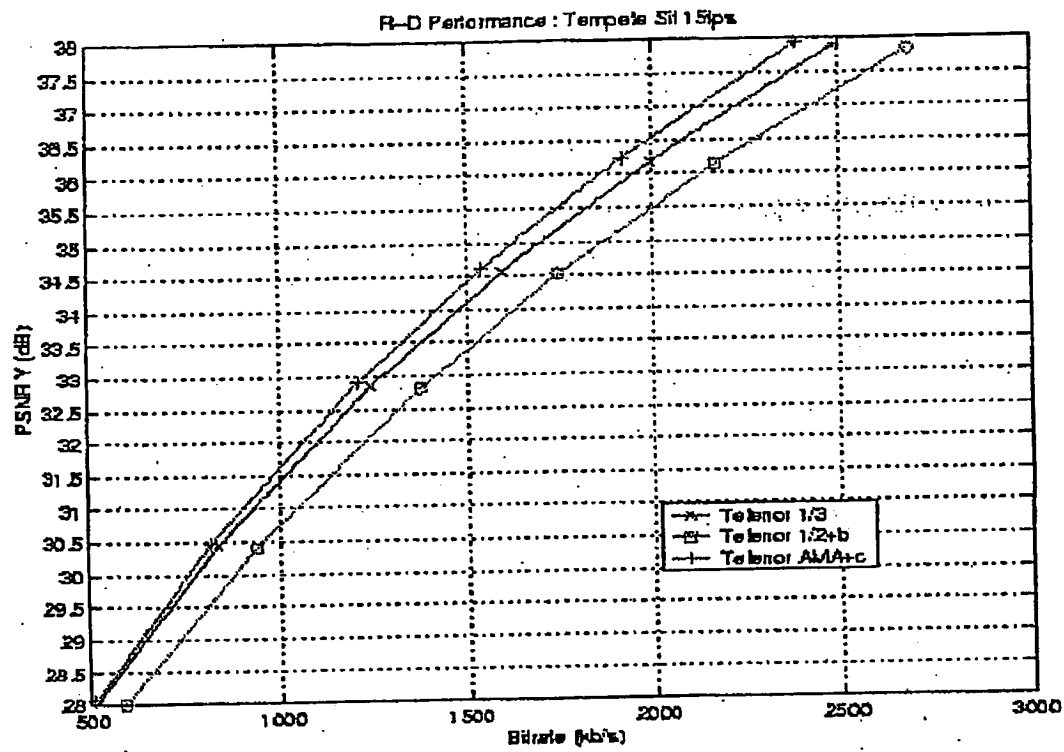


FIG.14

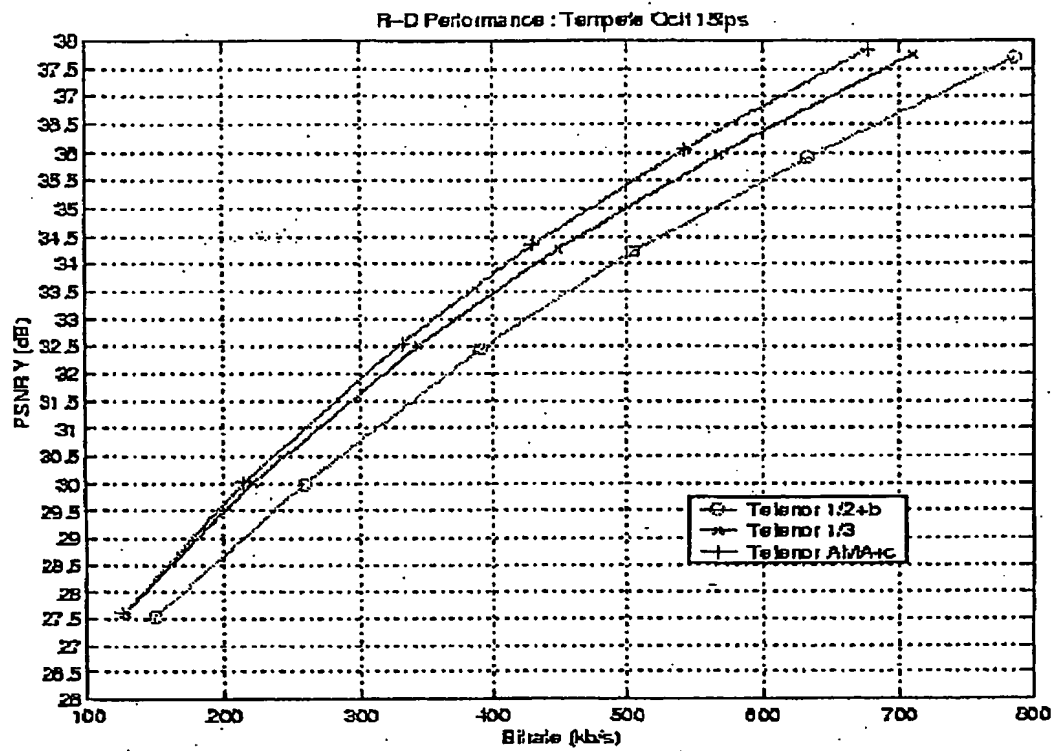


FIG.15

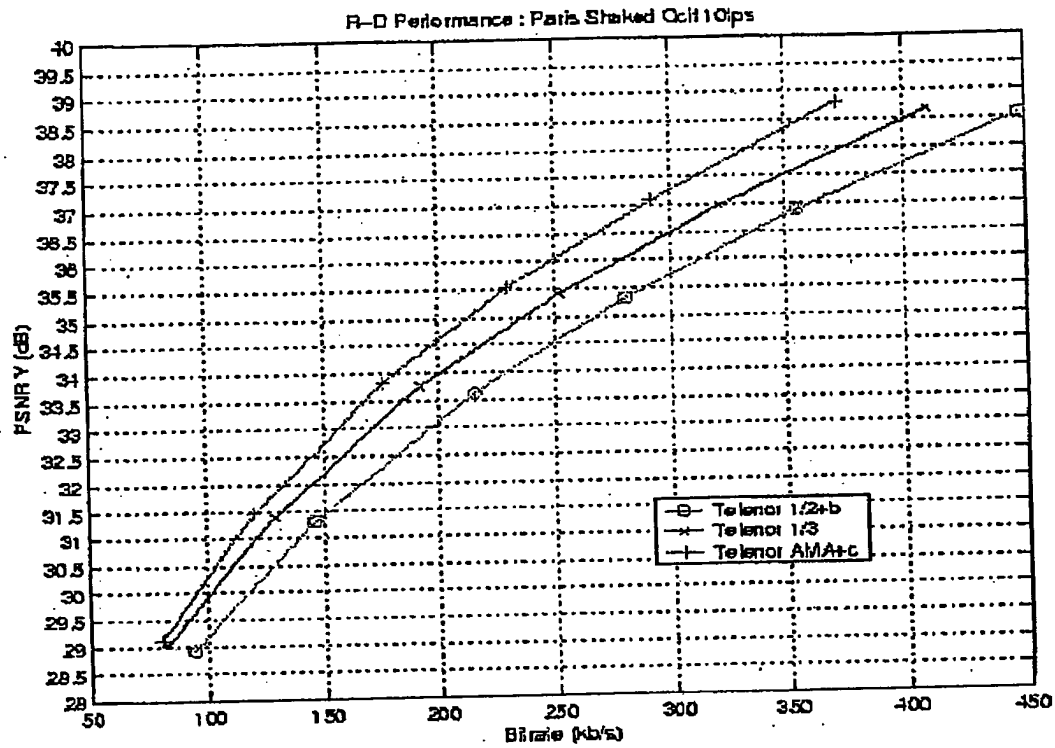




FIG.16

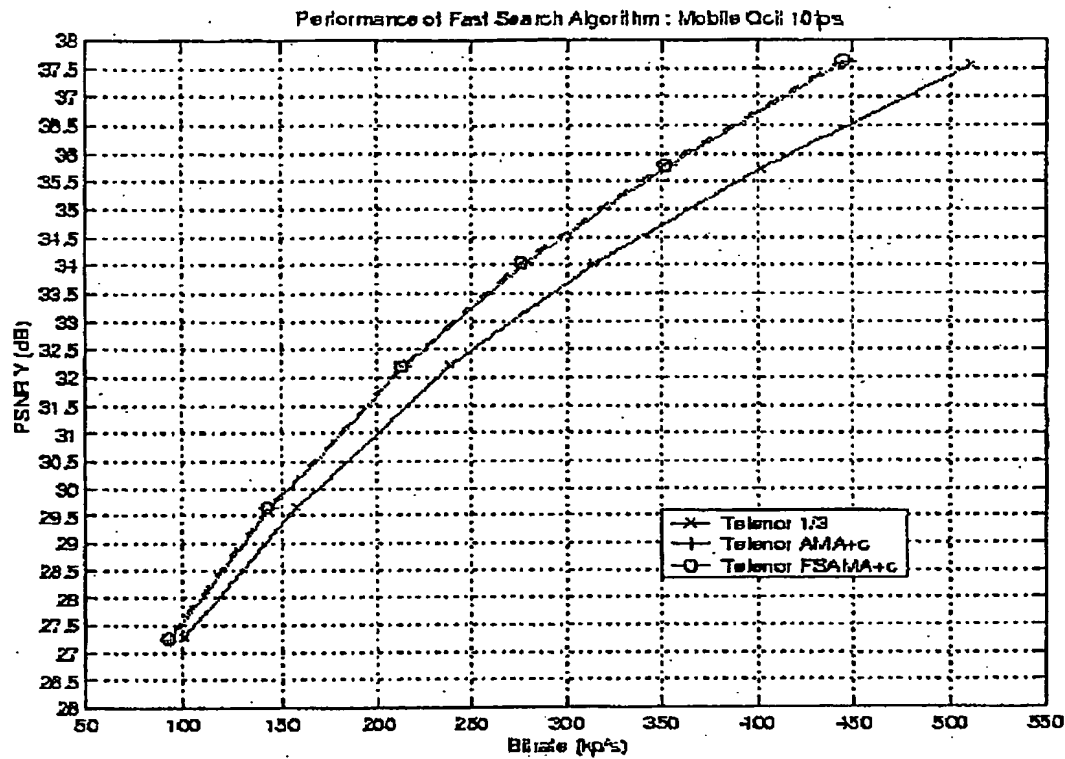


FIG.17

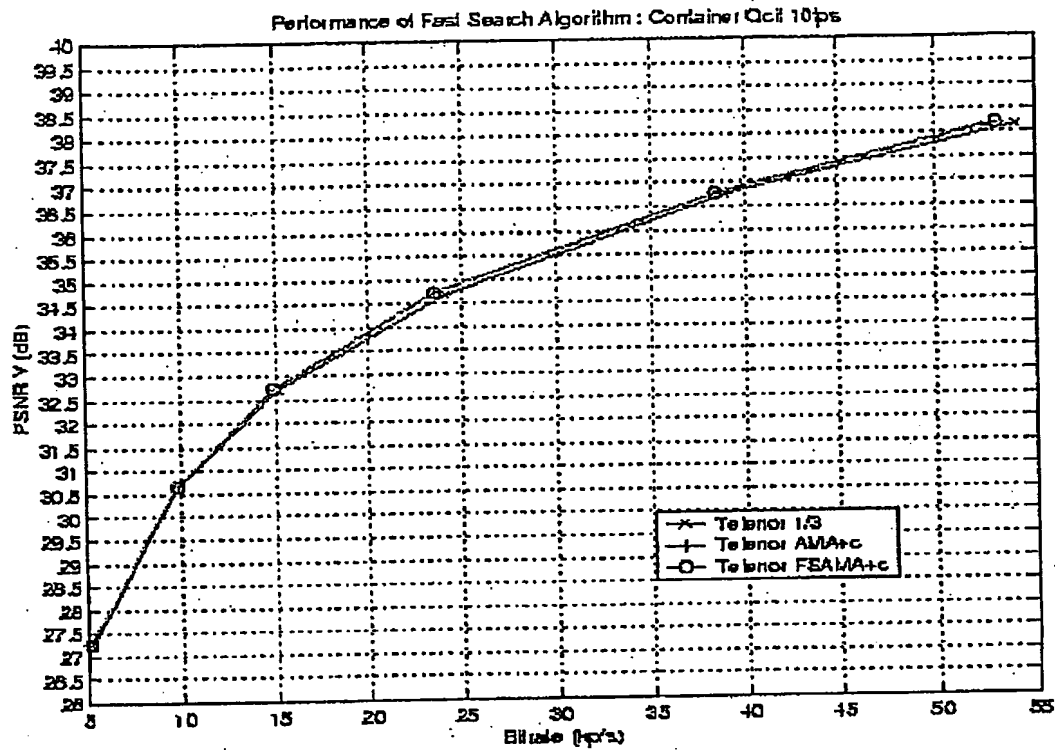
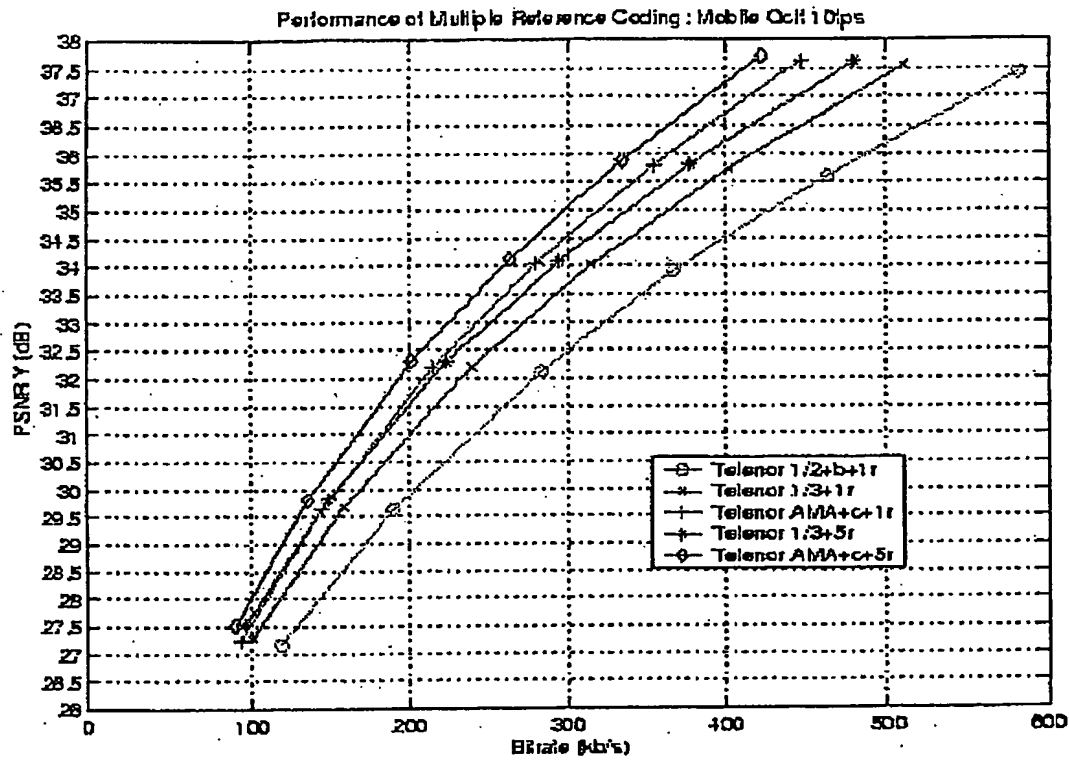
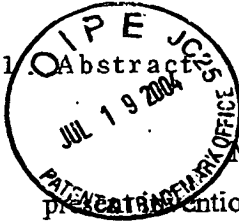


FIG.18





Methods for motion estimation with adaptive motion accuracy of the present invention include several techniques for computing motion vectors of high pixel accuracy with a minor increase in computation. One technique uses fast-search strategies in sub-pixel space that smartly searches for the best motion vectors. An alternate technique estimates high-accurate motion vectors using different interpolation filters at different stages in order to reduce computational complexity. Yet another technique uses rate-distortion criteria that adapts according to the different motion accuracies to determine both the best motion vectors and the best motion accuracies. Still another technique uses a VLC table that is interpreted differently at different coding units, according to the associated motion vector accuracy.

## 2. Representative Drawing

Fig.6

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**